



UNIVERSITY OF CALIFORNIA, BERKELEY
COLLEGE OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE
AND MATHEMATICS

Binary Search Tree (BST)

What is a binary search tree?

A *binary search tree* is a rooted, *binary tree* that is *ordered*.

A *binary tree*

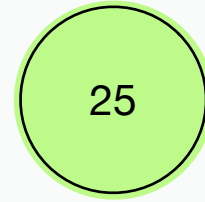


What is a binary search tree?

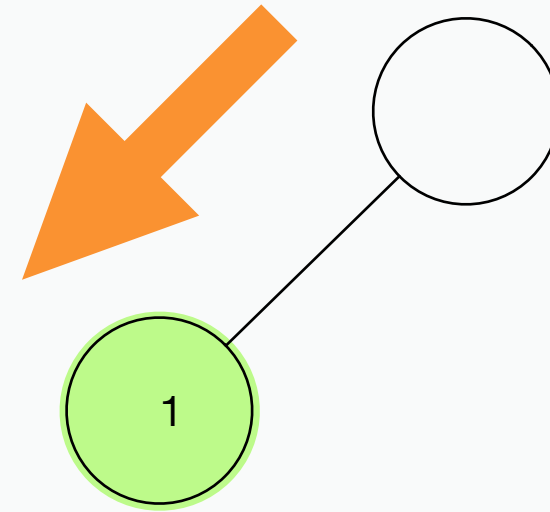
In a binary search tree:

- any values that appear in the tree appear exactly once (no duplicate values), and
- in-order traversal yields a sorted list of all values in the tree.

Constructing a binary search tree

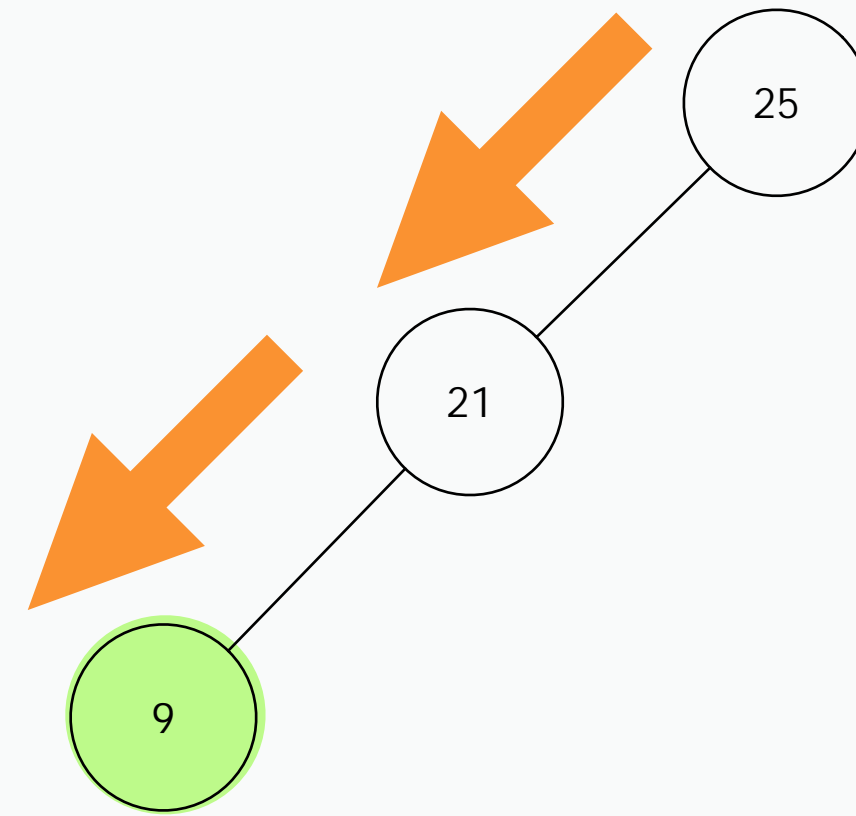


Constructing a binary search tree



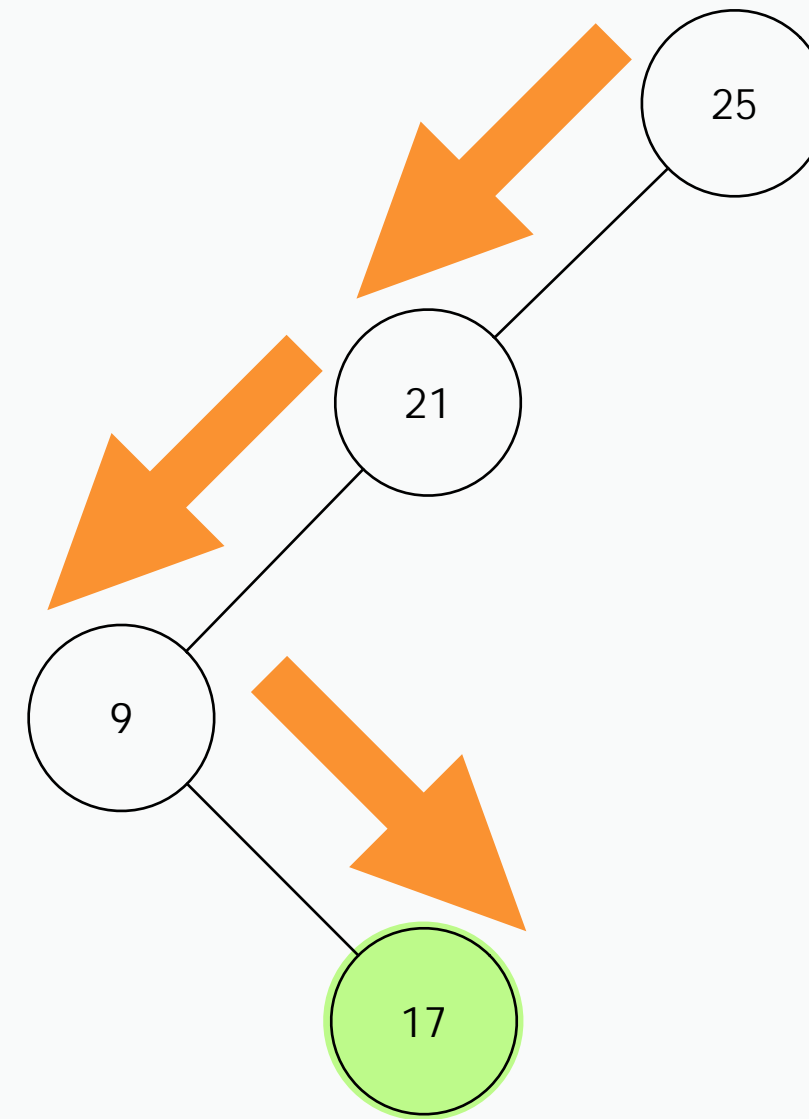
25, 21

Constructing a binary search tree



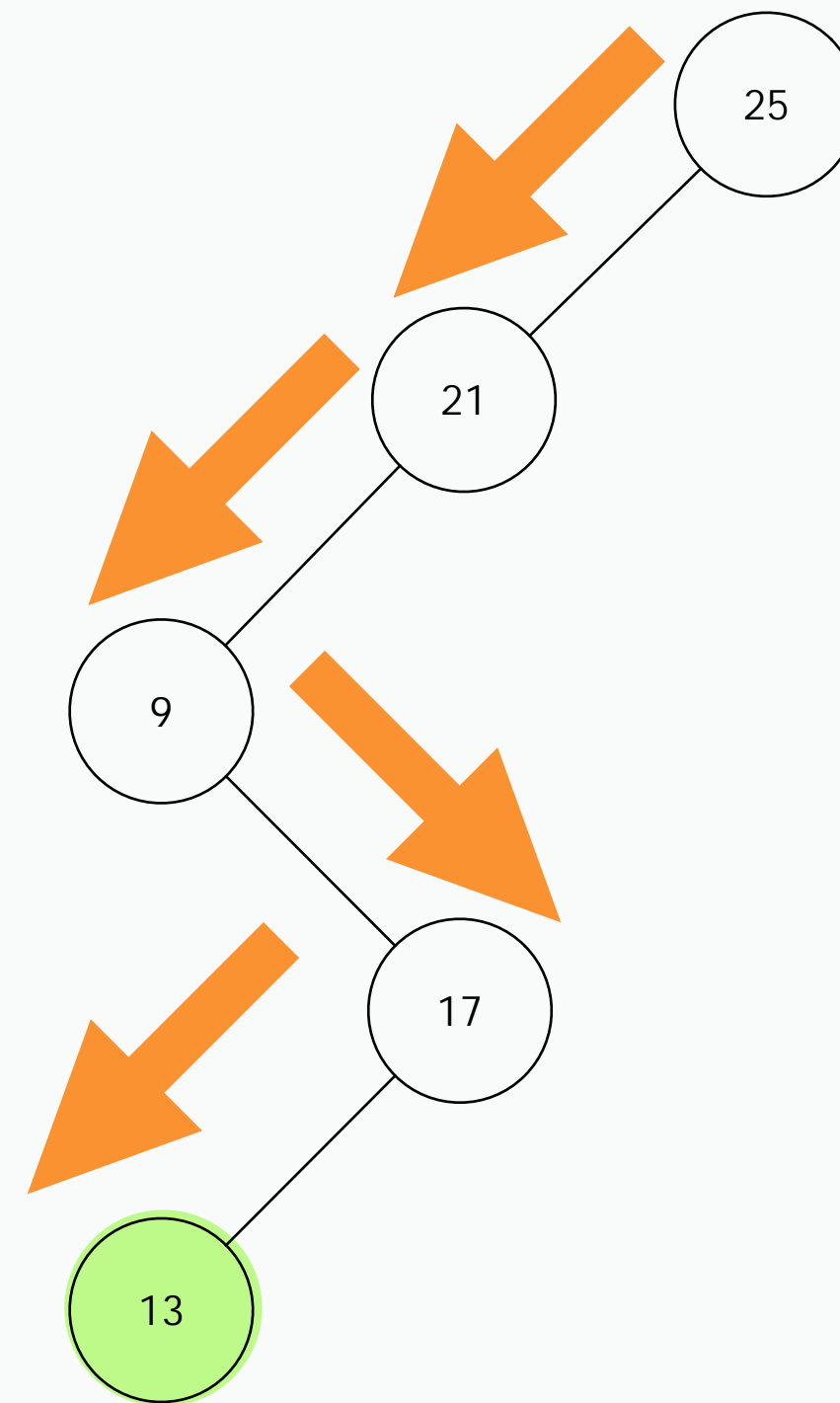
25, 21, 9

Constructing a binary search tree



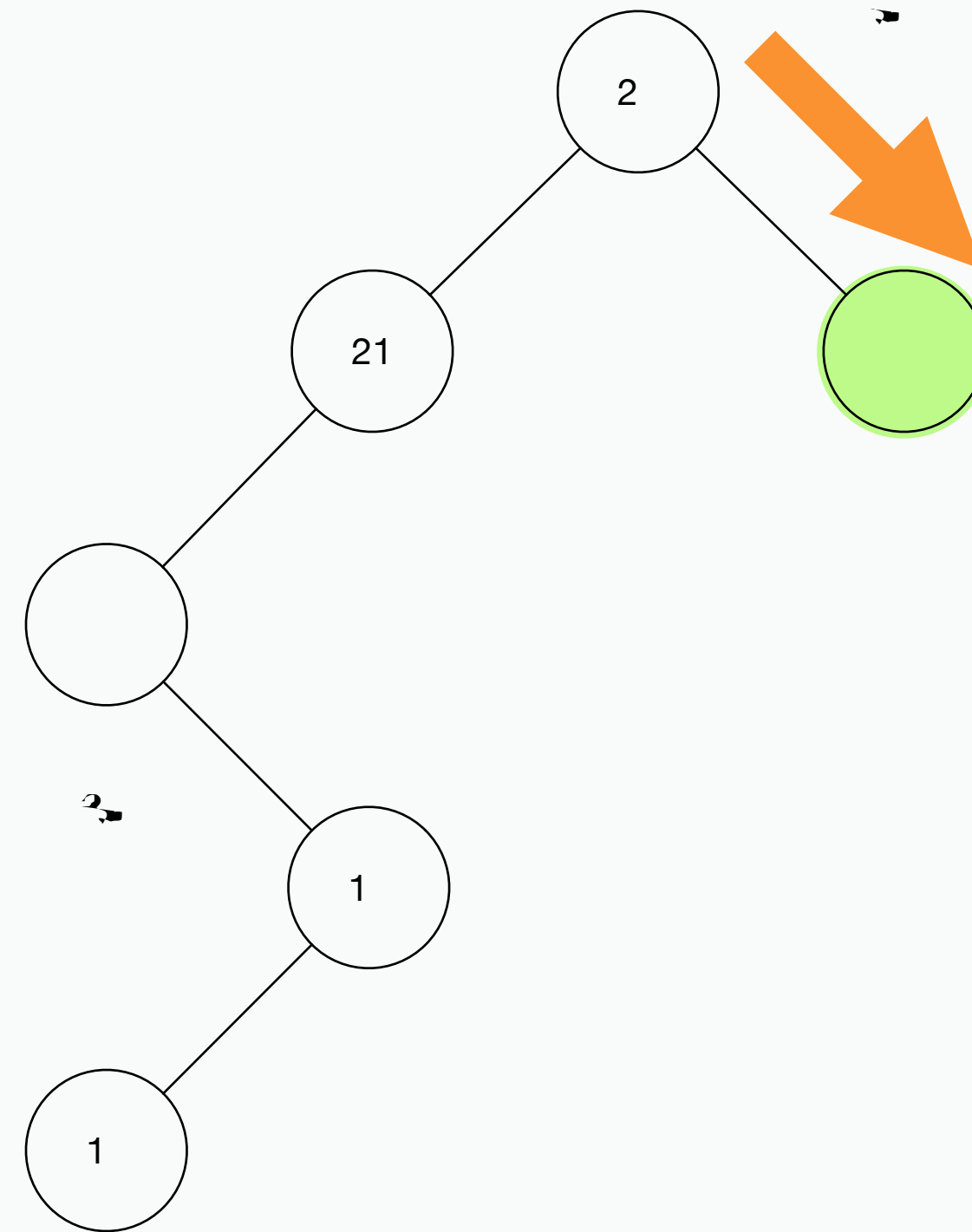
25, 21, 9, 17

Constructing a binary search tree



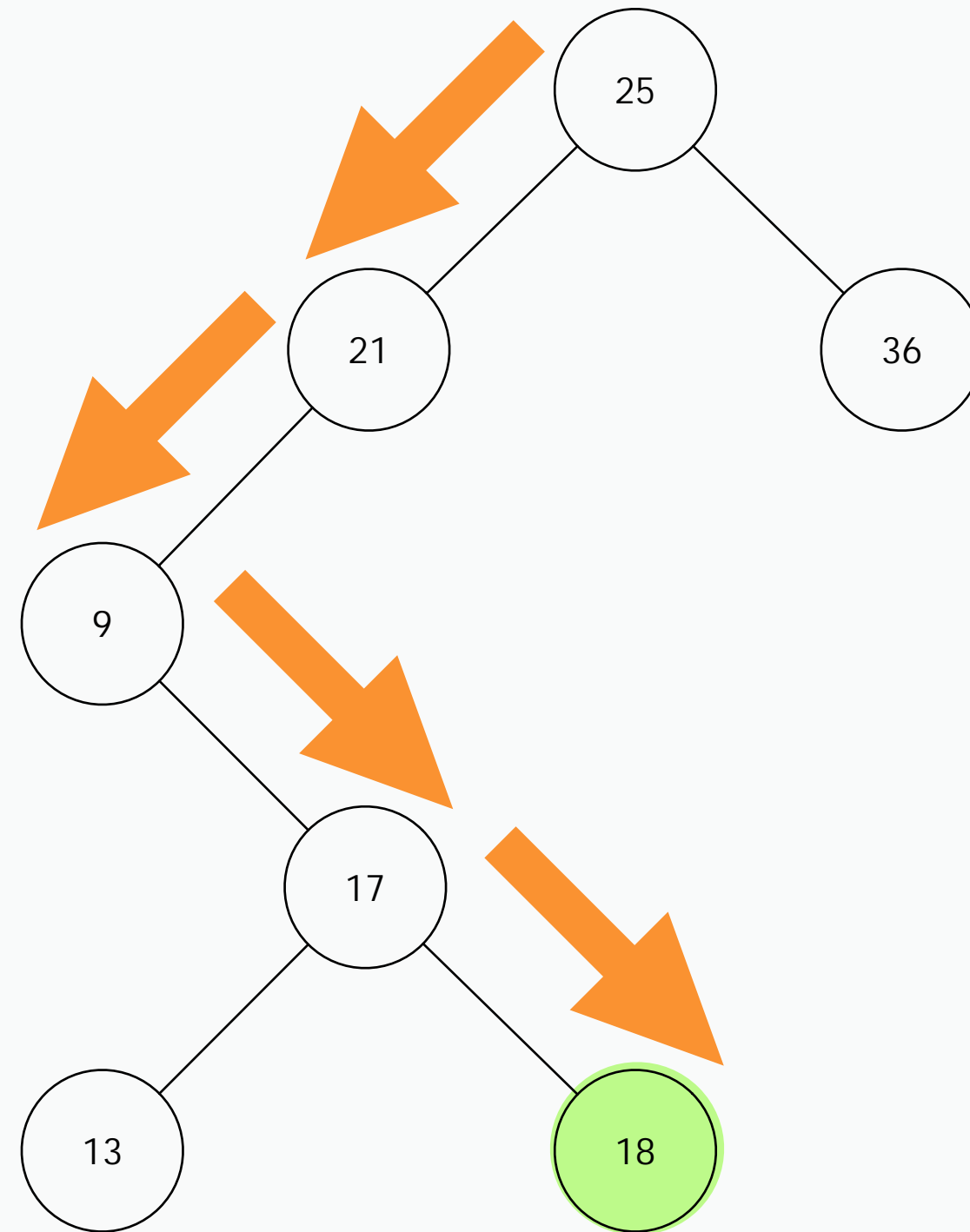
25, 21, 9, 17, 13

Constructing a binary search tree



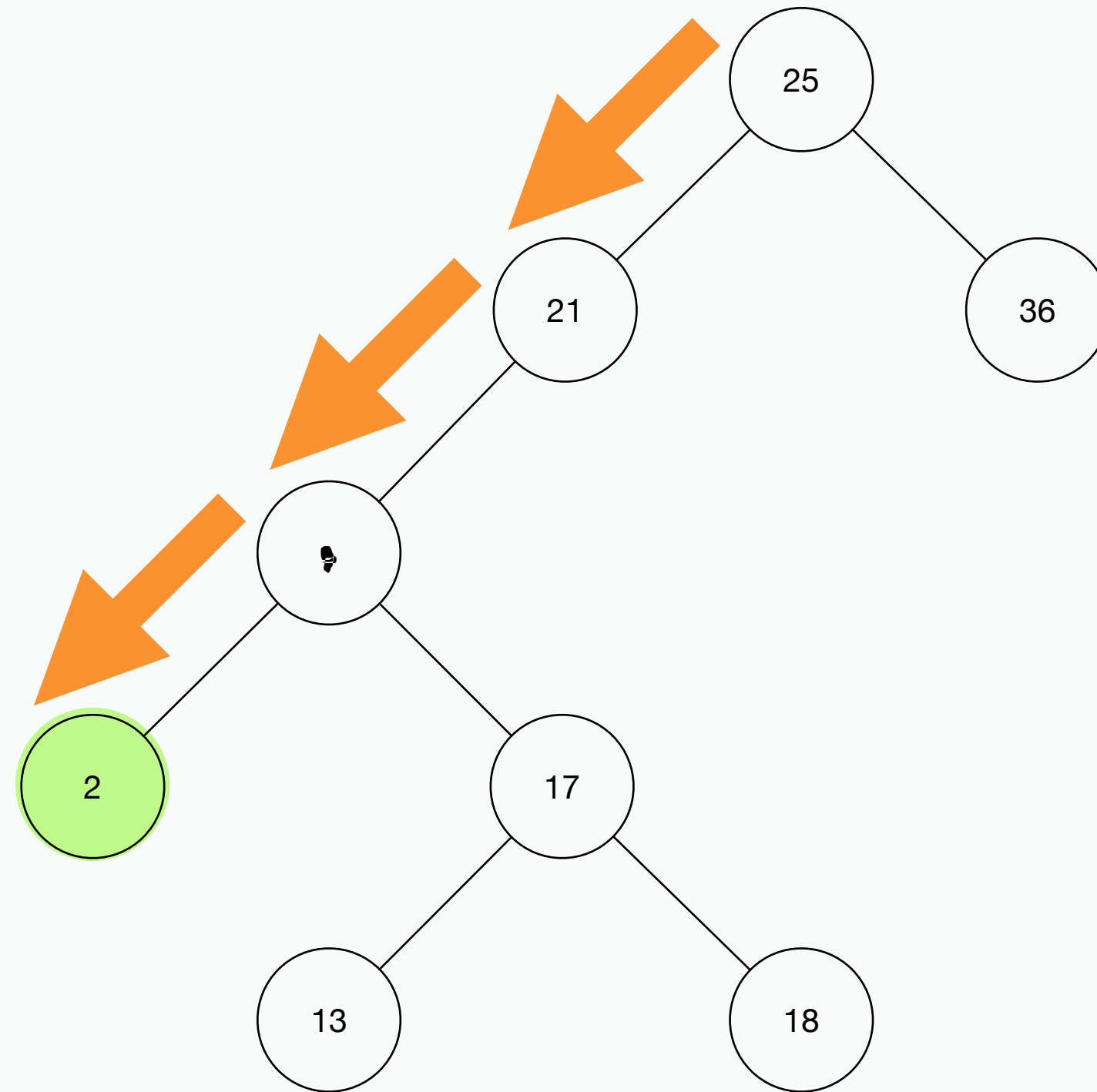
25, 21, 9, 17, 13, 36

Constructing a binary search tree



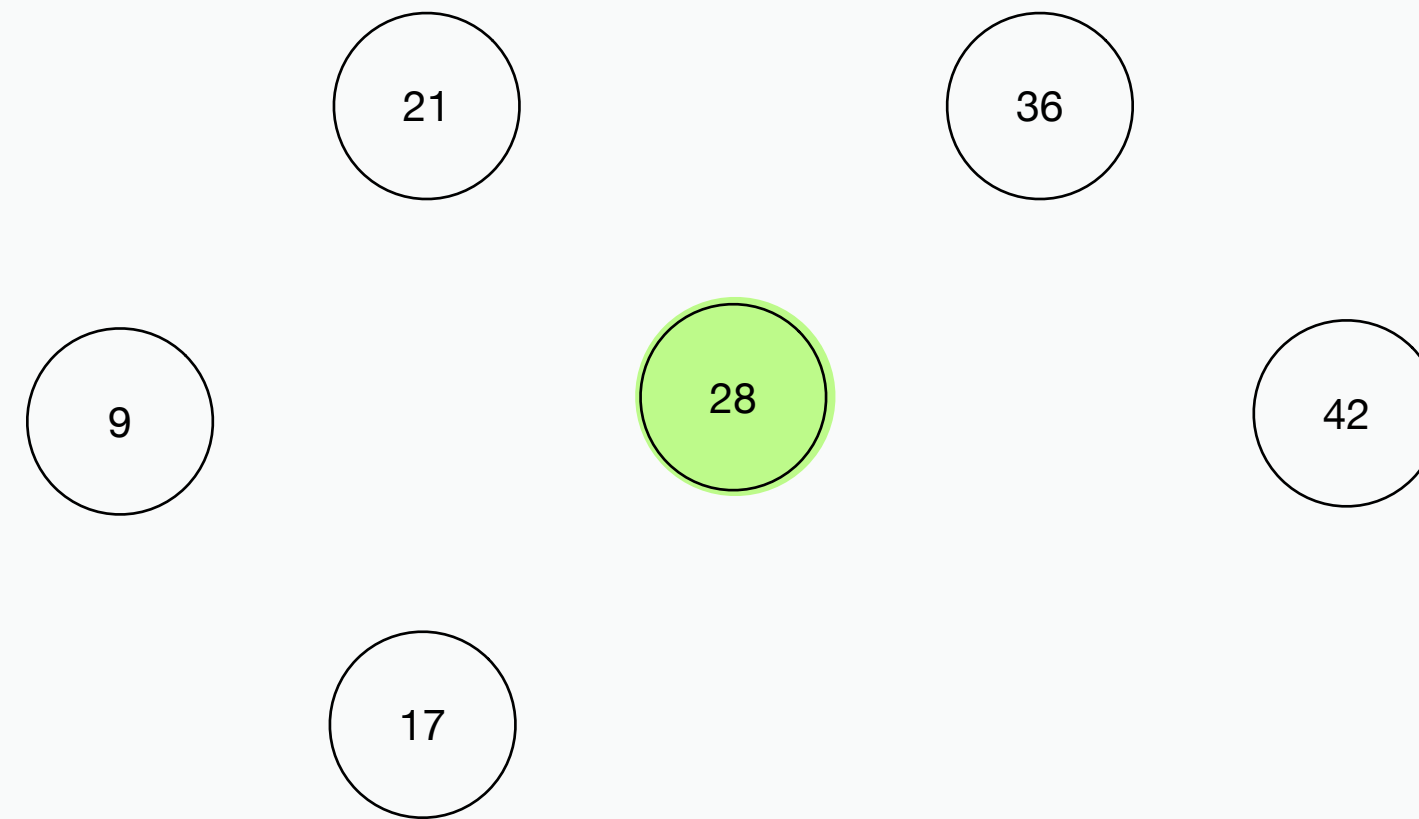
25, 21, 9, 17, 13, 36, 18

Constructing a binary search tree



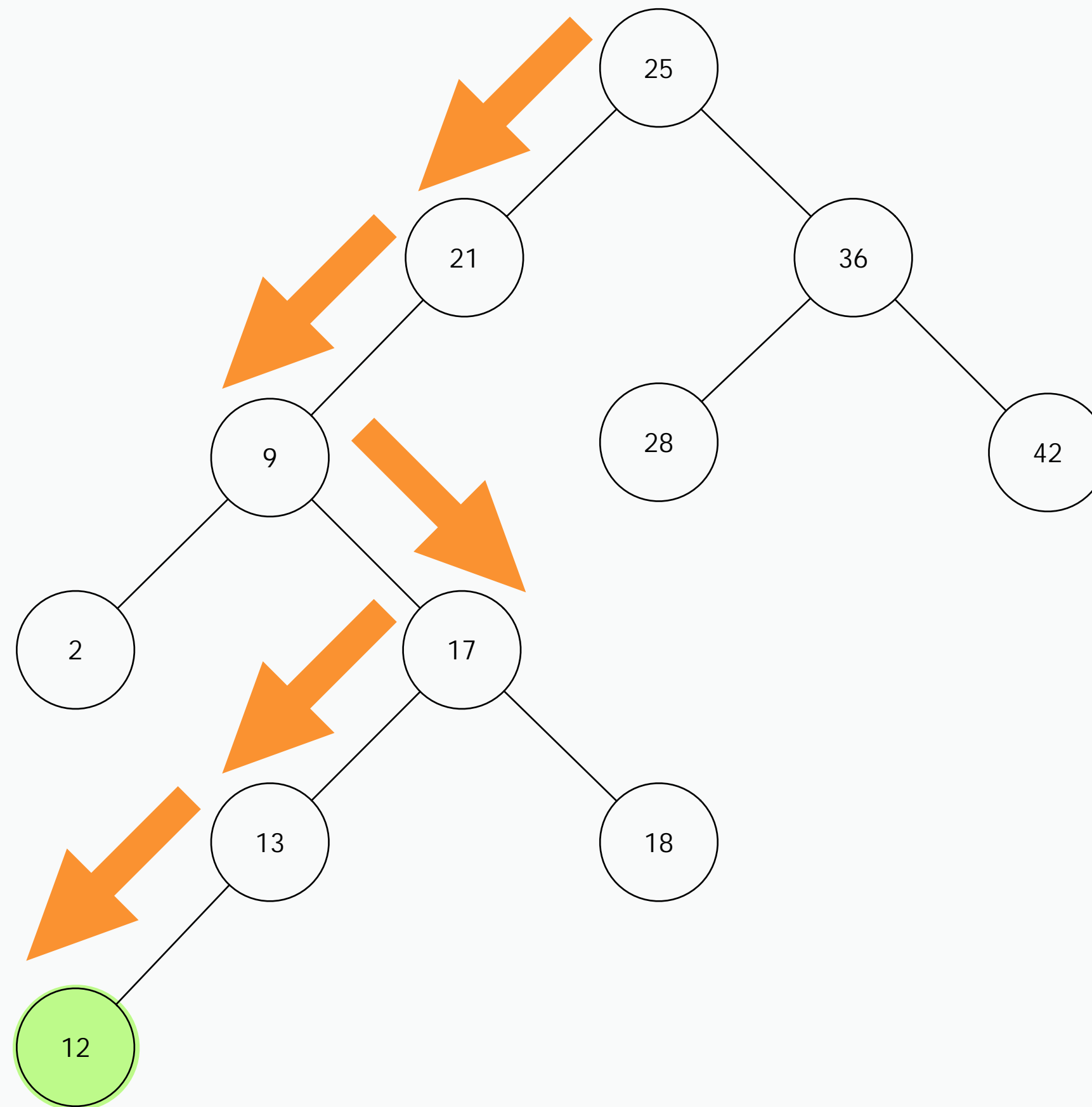
25, 21, 9, 17, 13, 36, 18, 2

Constructing a binary search tree



25, 21, 9, 17, 13, 36, 18, 2, 42, 28

Constructing a binary search tree



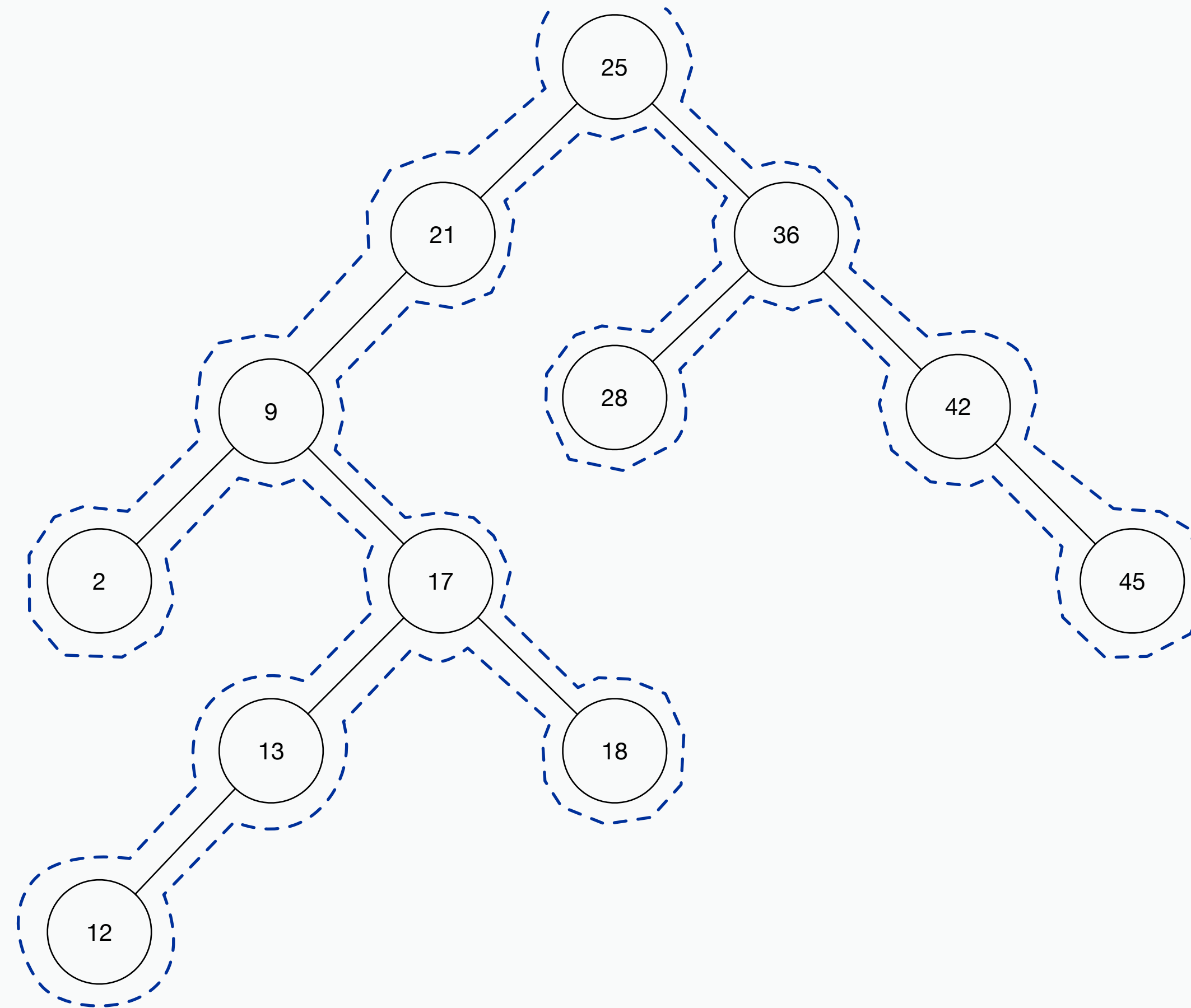
25, 21, 9, 17, 13, 36, 18, 2, 42, 28, 12

Constructing a binary search tree

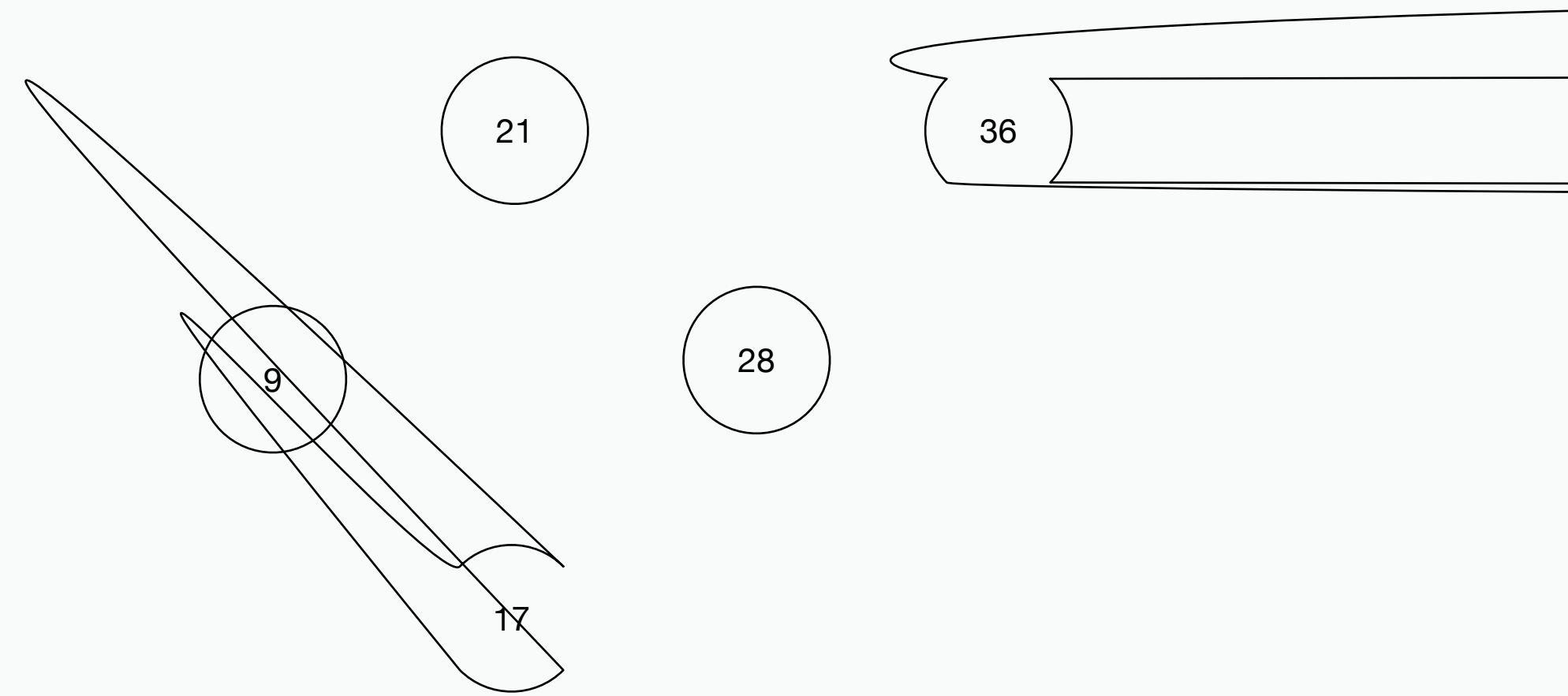


25, 21, 9, 17, 13, 36, 18, 2, 42, 28, 12, 45

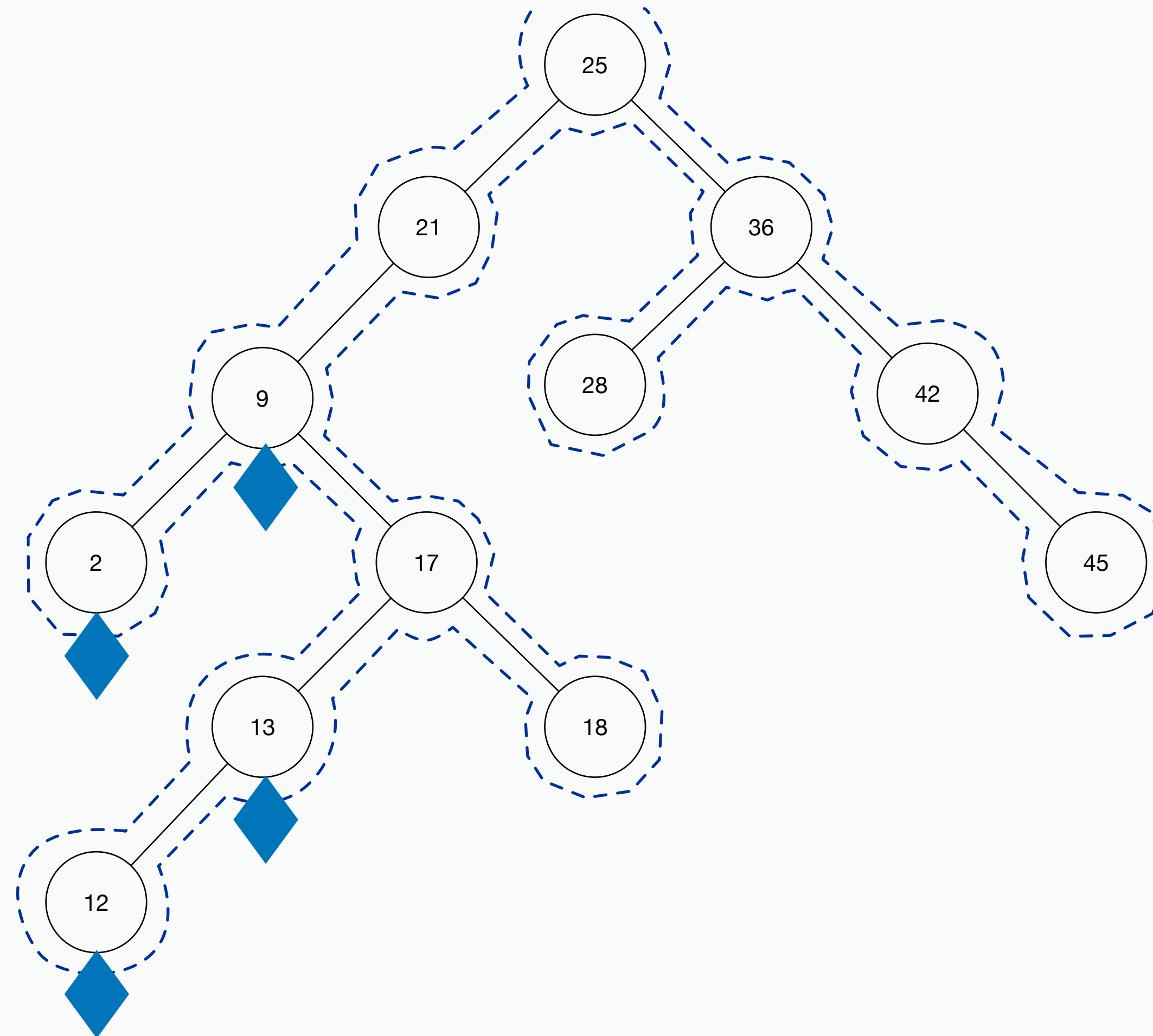
Constructing a binary search tree



Constructing a binary search tree

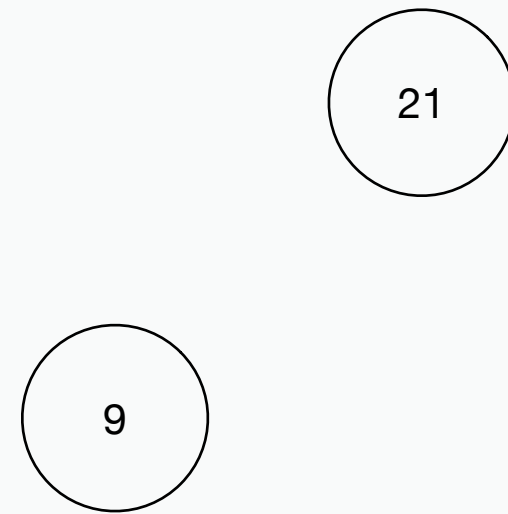


Constructing a binary search tree

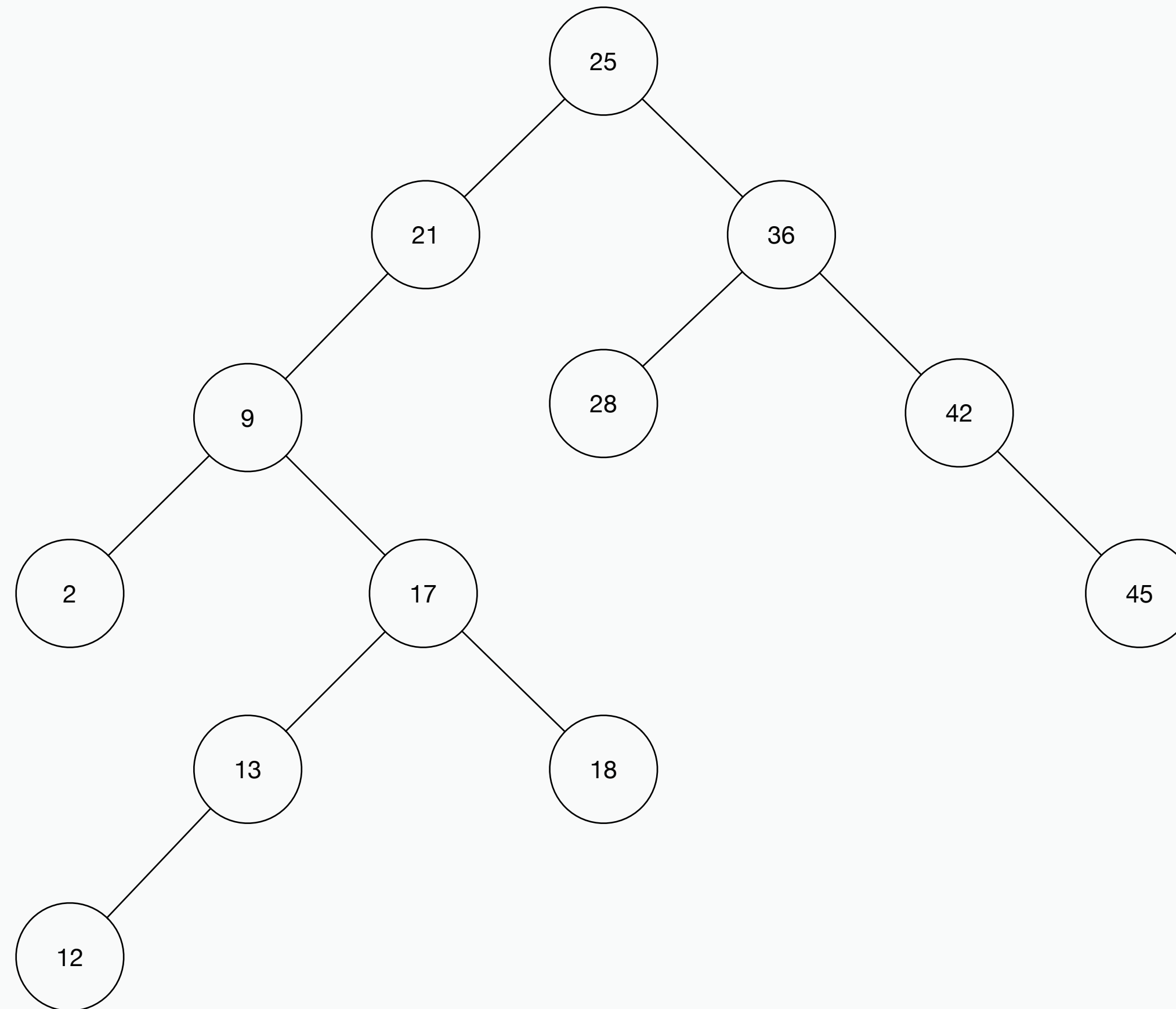


2, 9, 12, 13

Constructing a binary search tree



Constructing a binary search tree

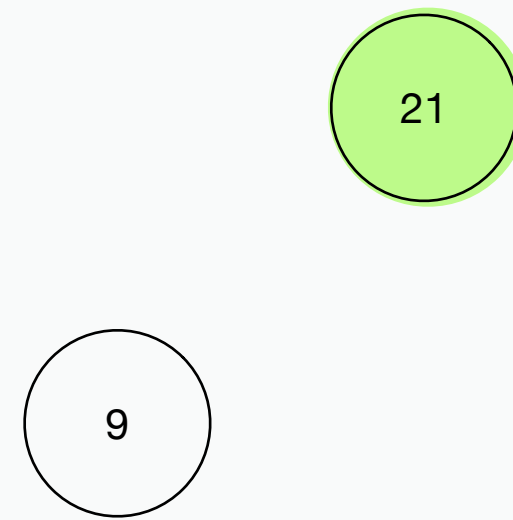


Constructing a binary search tree

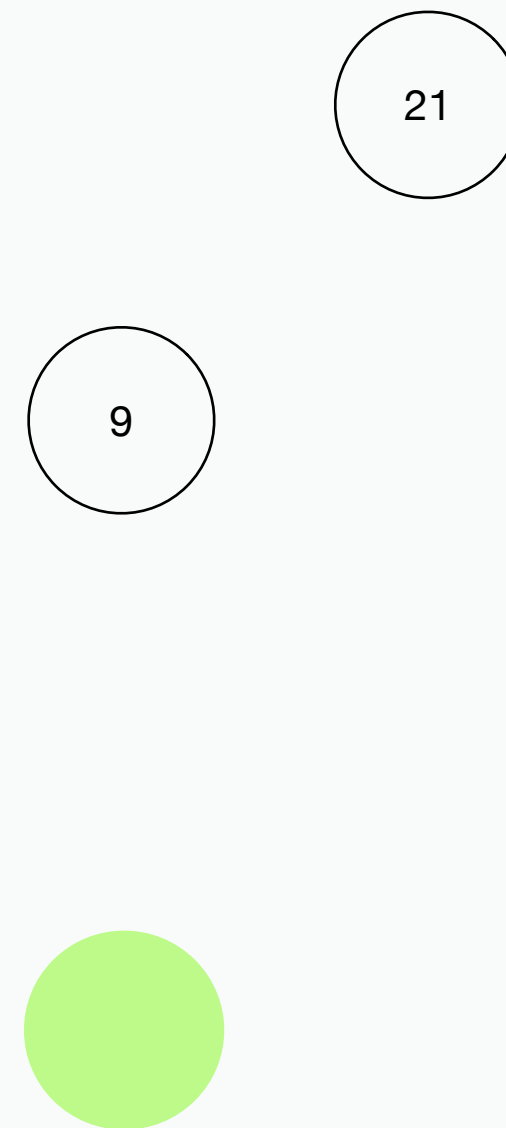
9

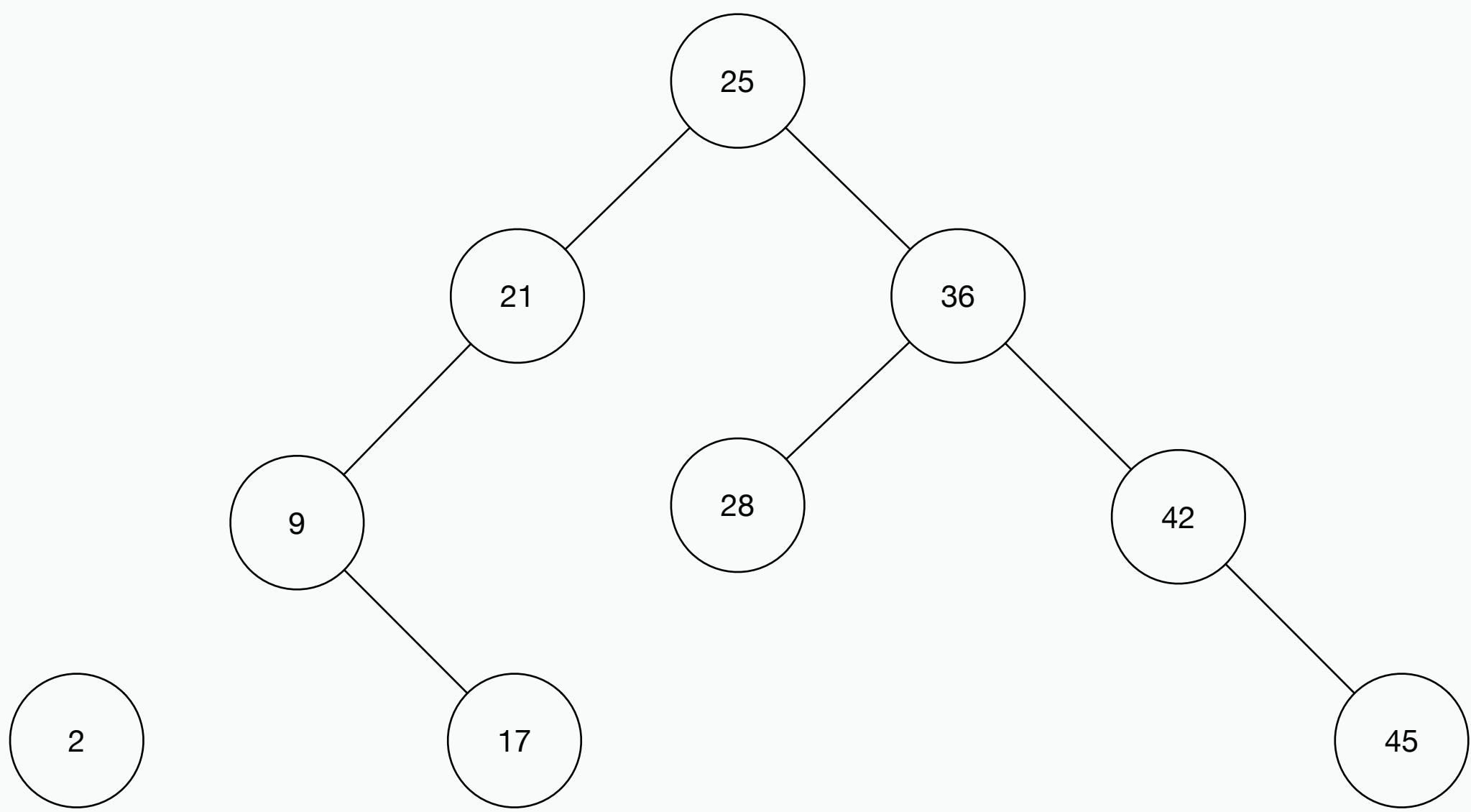
21

Searching a binary search tree



Searching a binary search tree



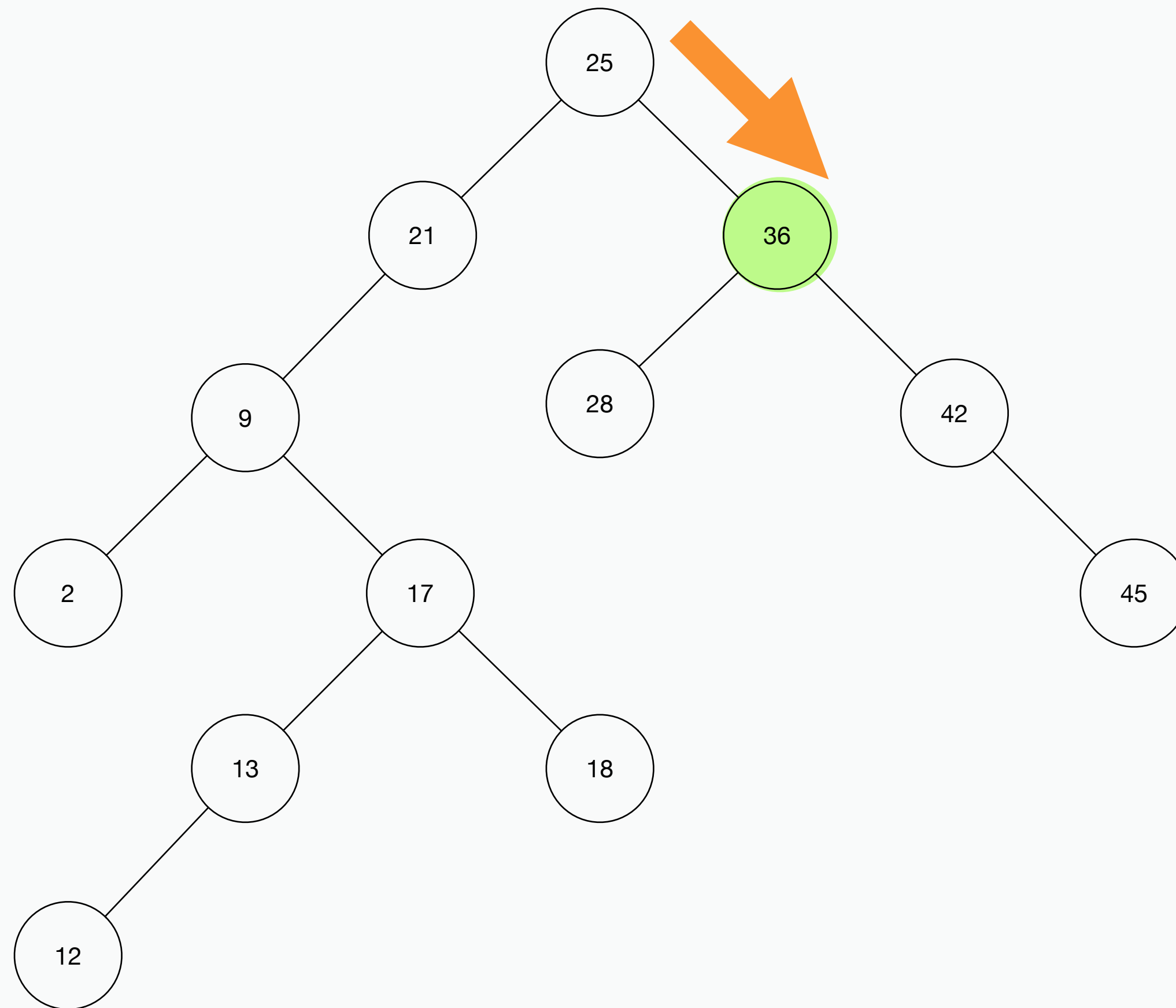


Searching a binary search tree

37 (not in tree)

25? $37 > 25$. Go right.

36? $37 > 36$. Go right.



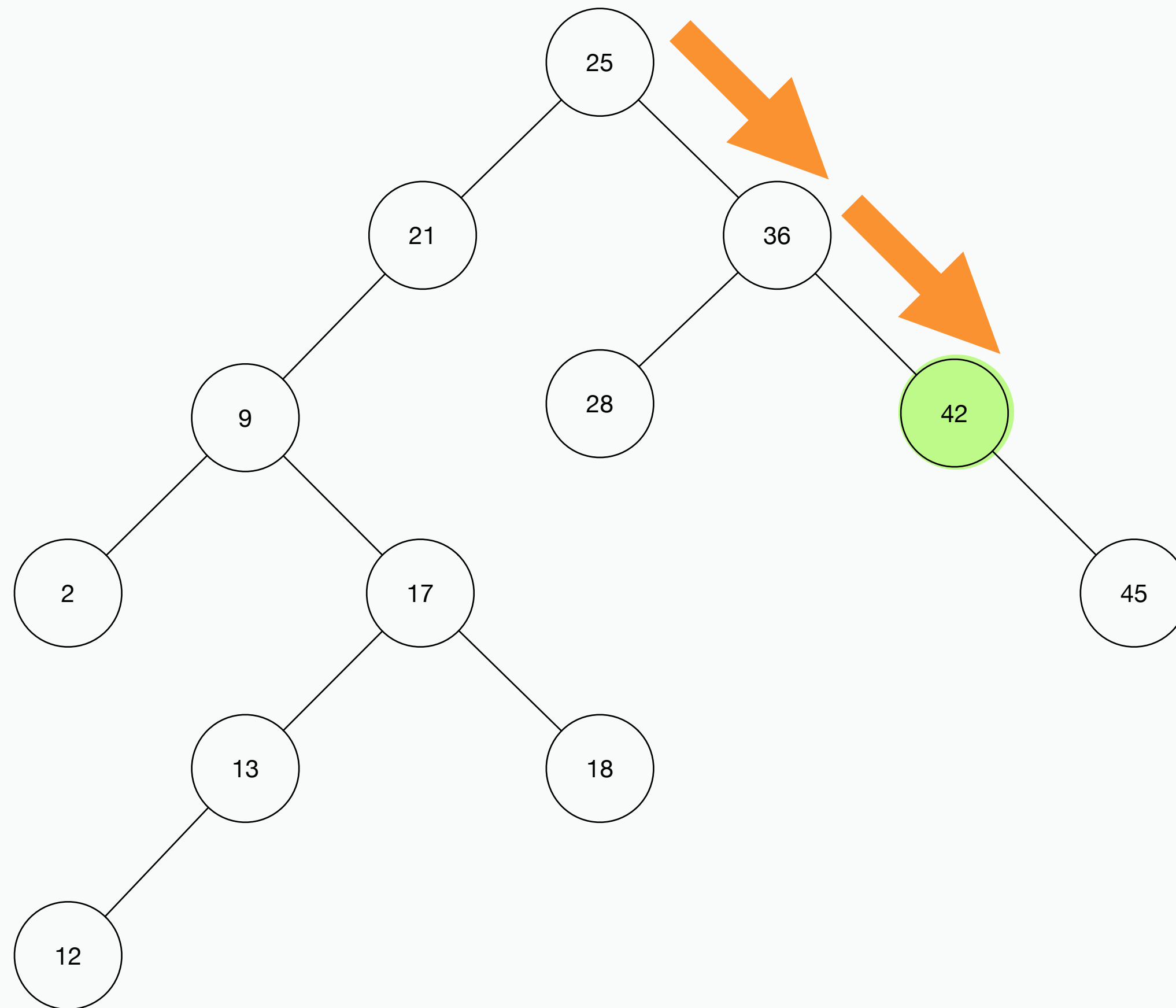
Searching a binary search tree

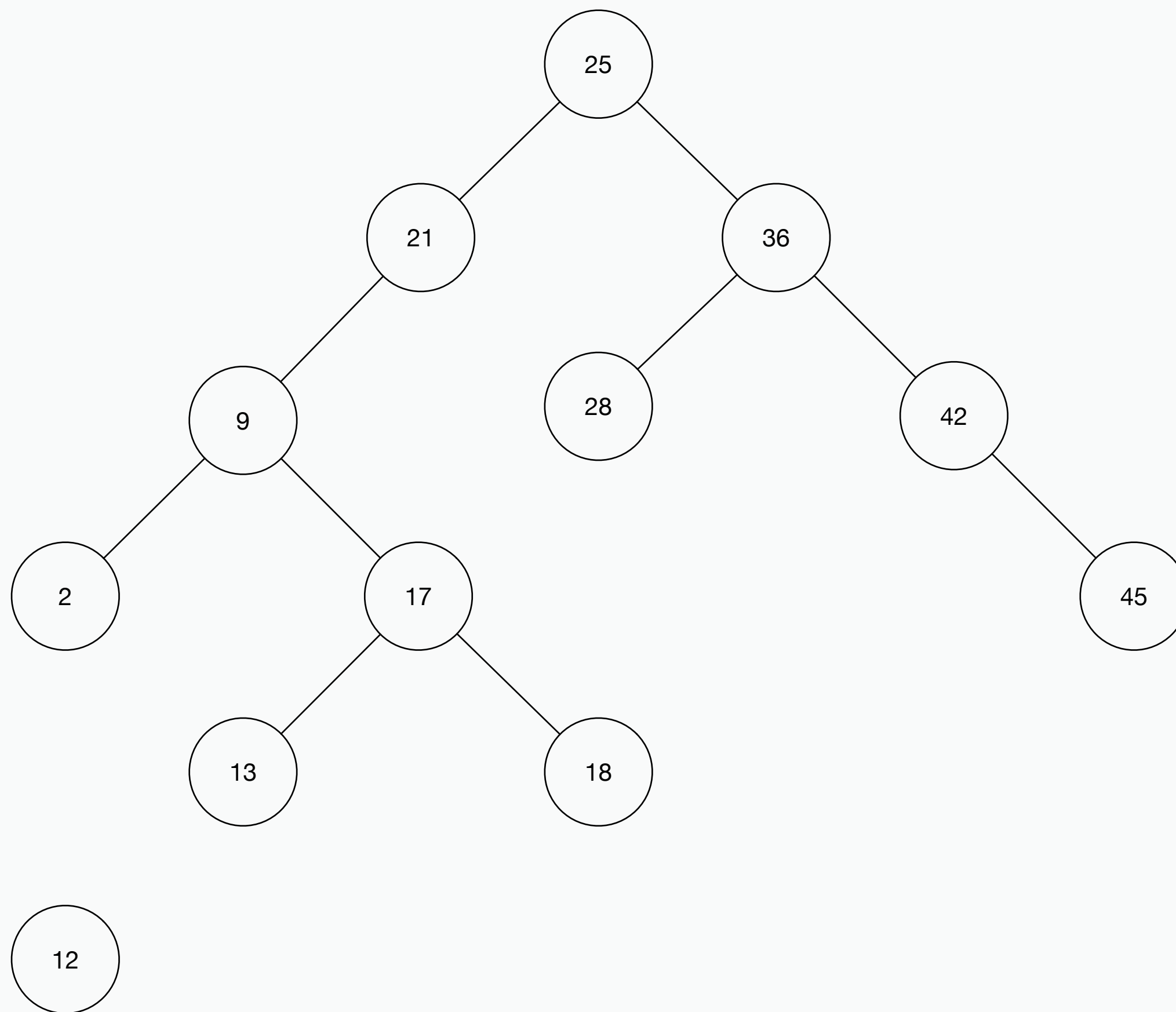
37 (not in tree)

25? $37 > 25$. Go right.

36? $37 > 36$. Go right.

42? $37 < 42$. Go left.





Complexity of search

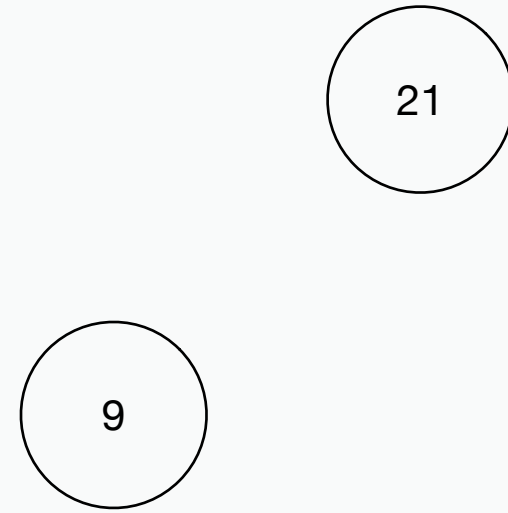
Deleting nodes in a BST

Can get a wee bit tricky / four cases

- Target node is a leaf. Delete the node.
- Target node has one child. Delete the node and replace it with its child.
- Target node has two children:
 - Target node's left child has no right child. Delete the node and replace it with its left child.
 - Target node's left child has a right child. From the target node's left child's right child, continue to probe down through the tree, following right children until you can proceed no further. Replace the target node with the node found by probing. If the node found by probing has a left child, replace that node with its left child.

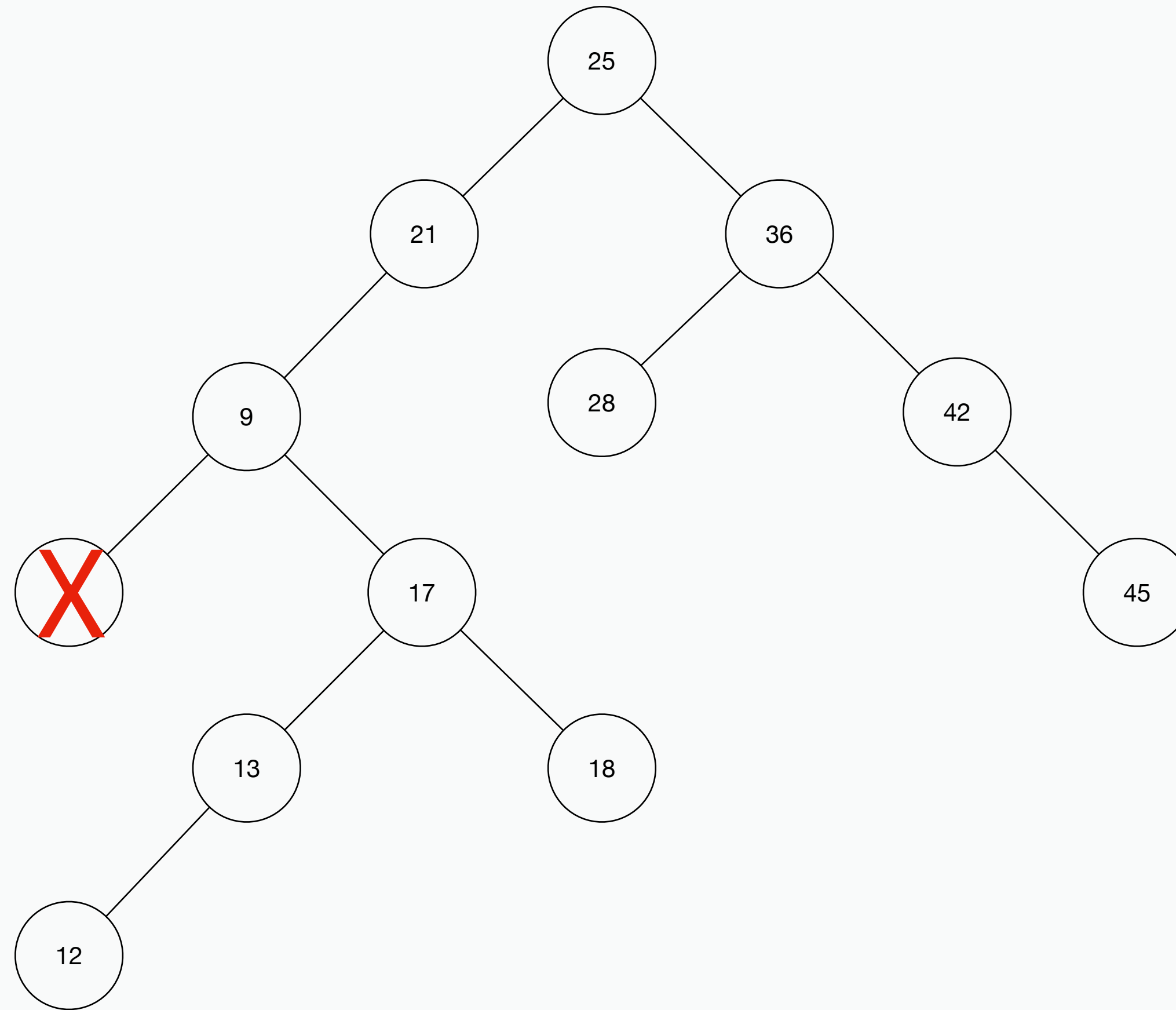
Deleting nodes in a BST

Case 1: Target node is a leaf.



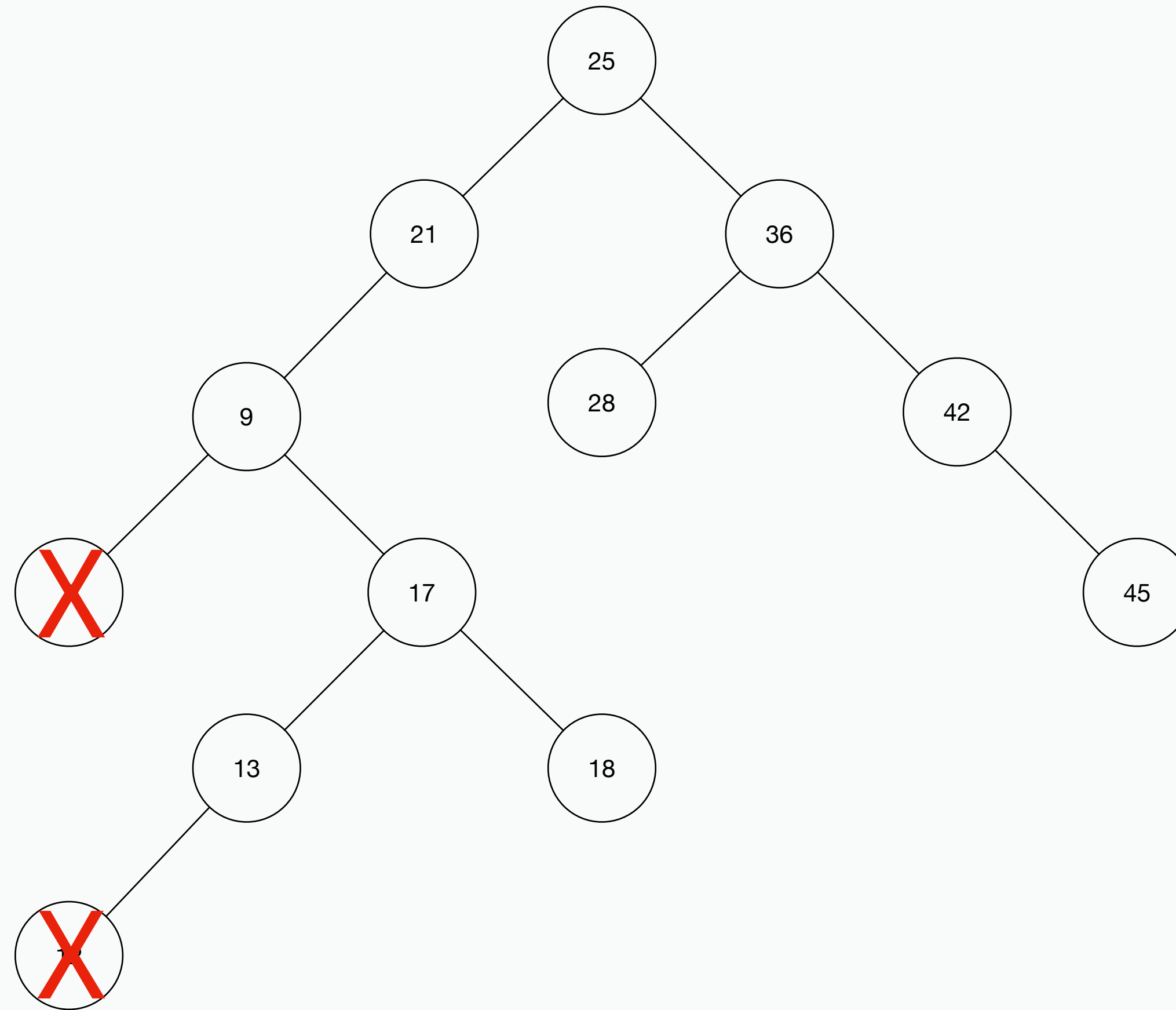
Deleting nodes in a BST

Case 1: Target node is a leaf.



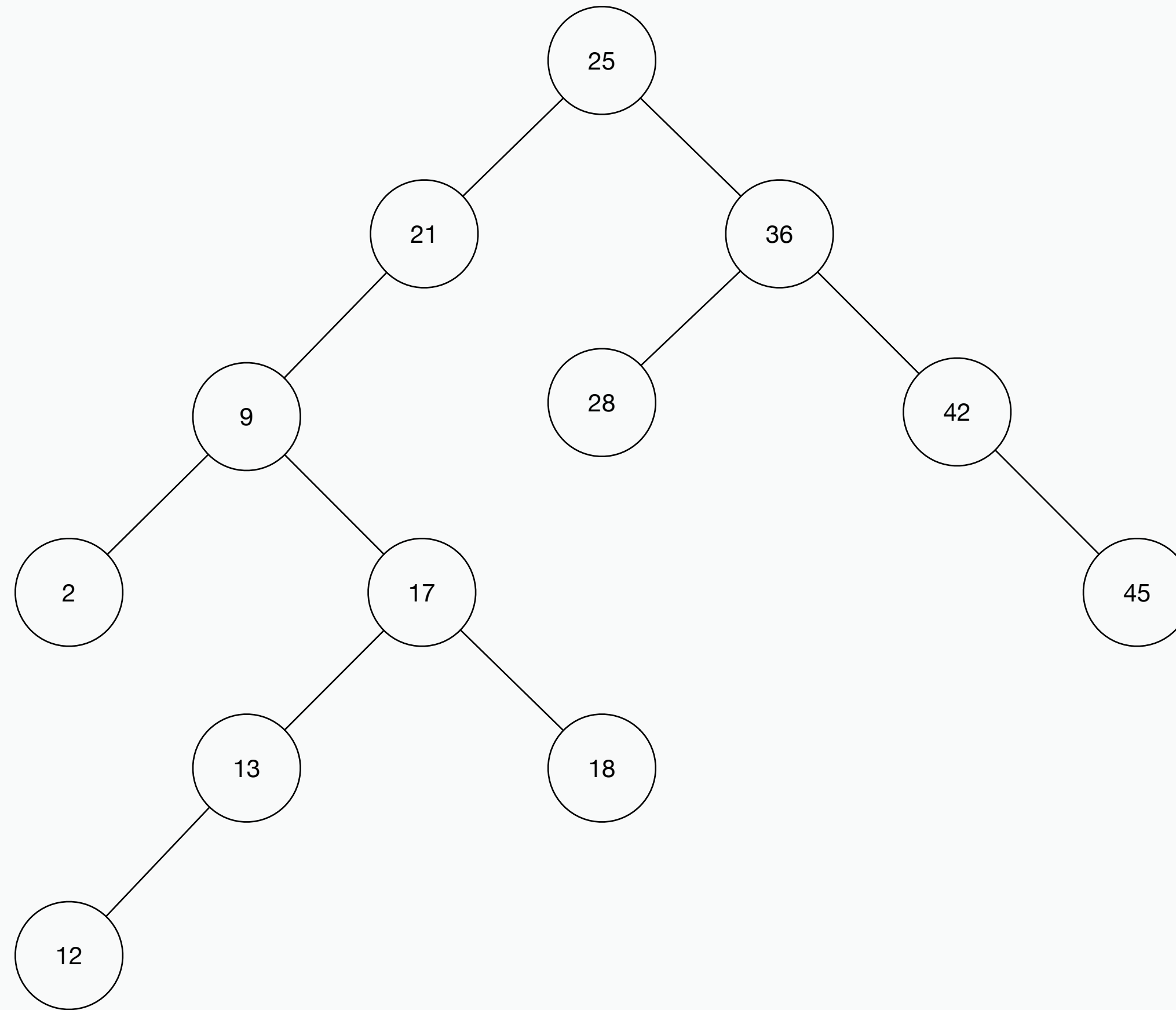
Deleting nodes in a BST

Case 1: Target node is a leaf.



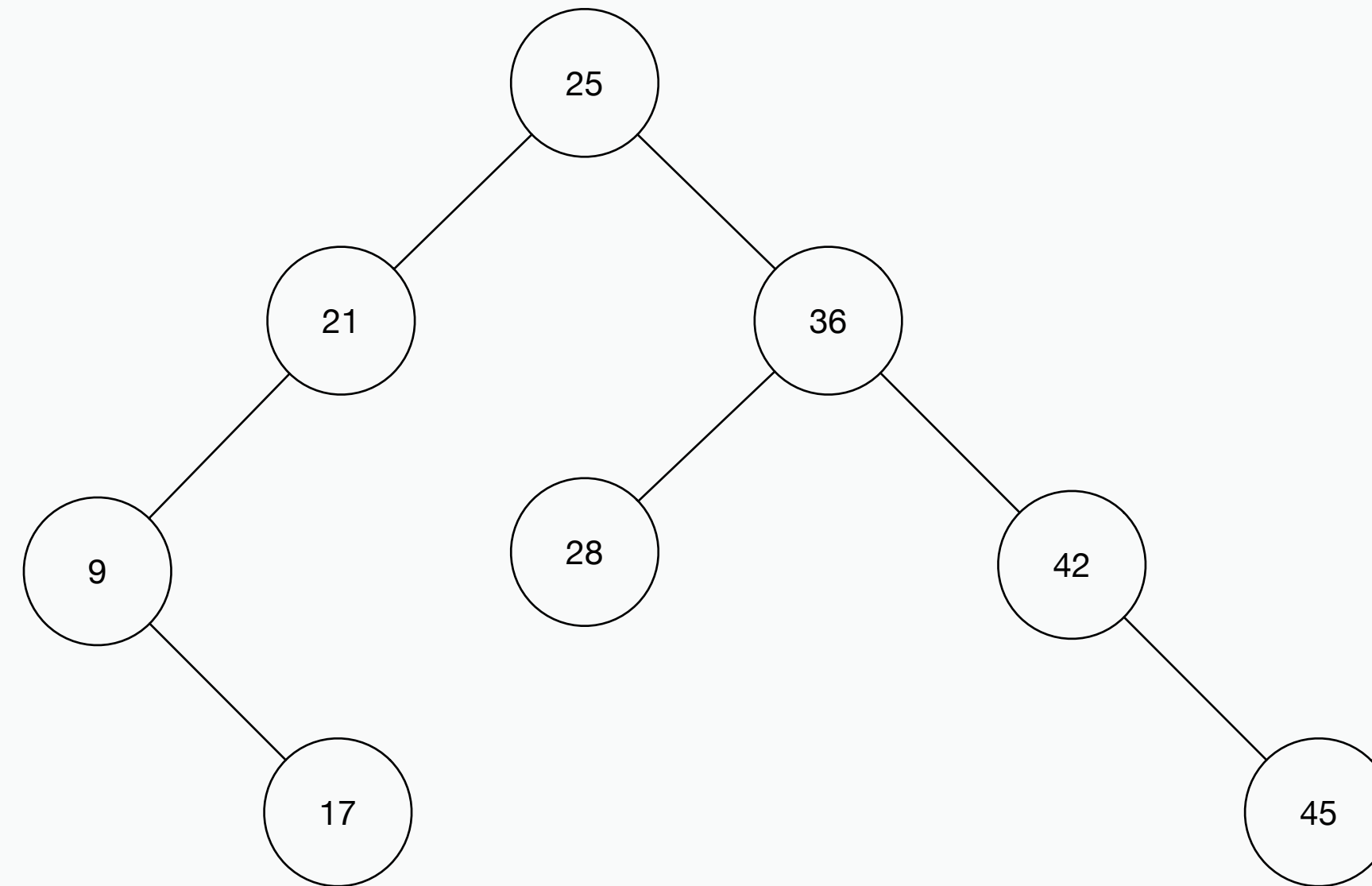
Deleting nodes in a BST

Case 1: Target node is a leaf.



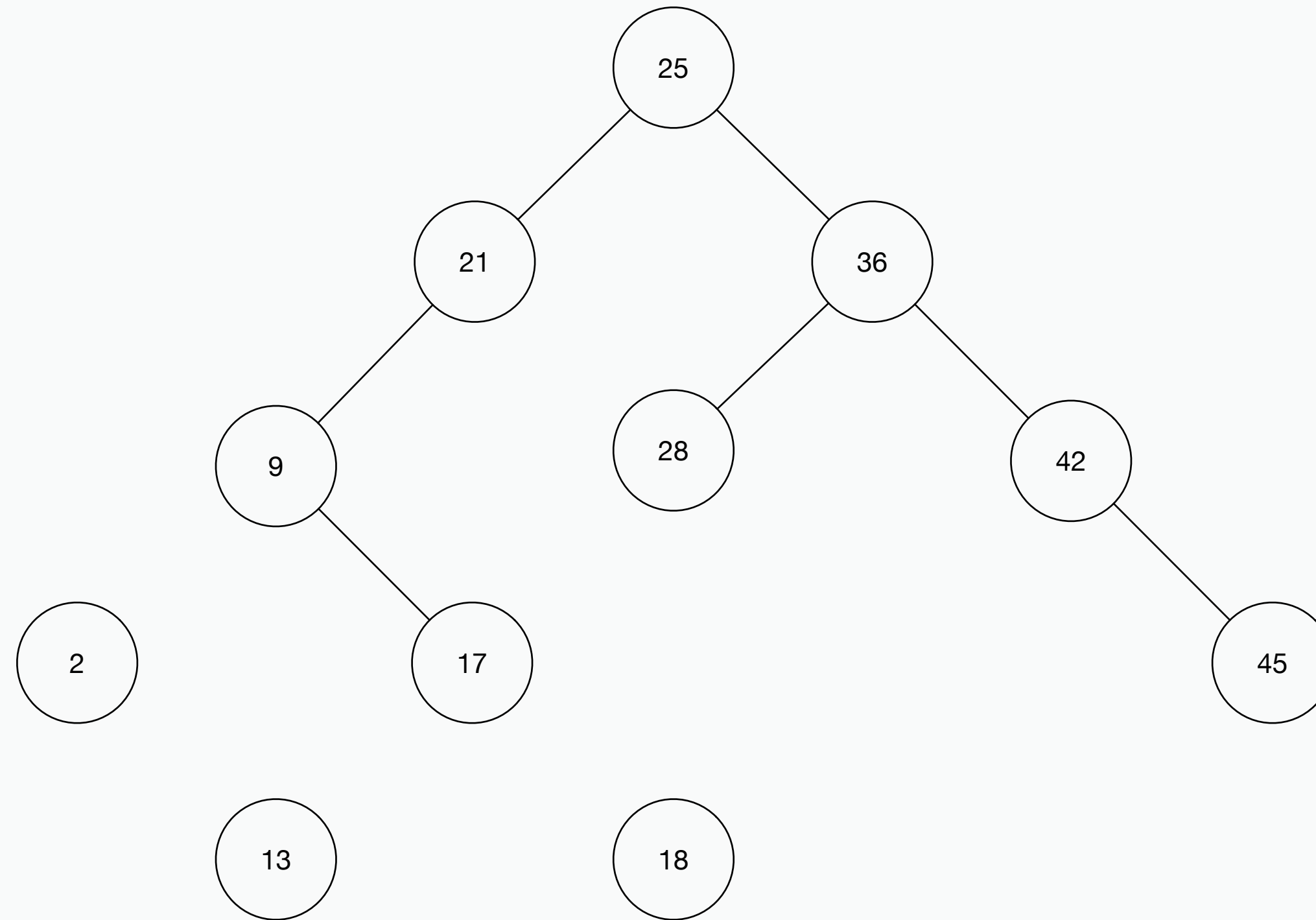
Deleting nodes in a BST

Case 2: Target node has one child.



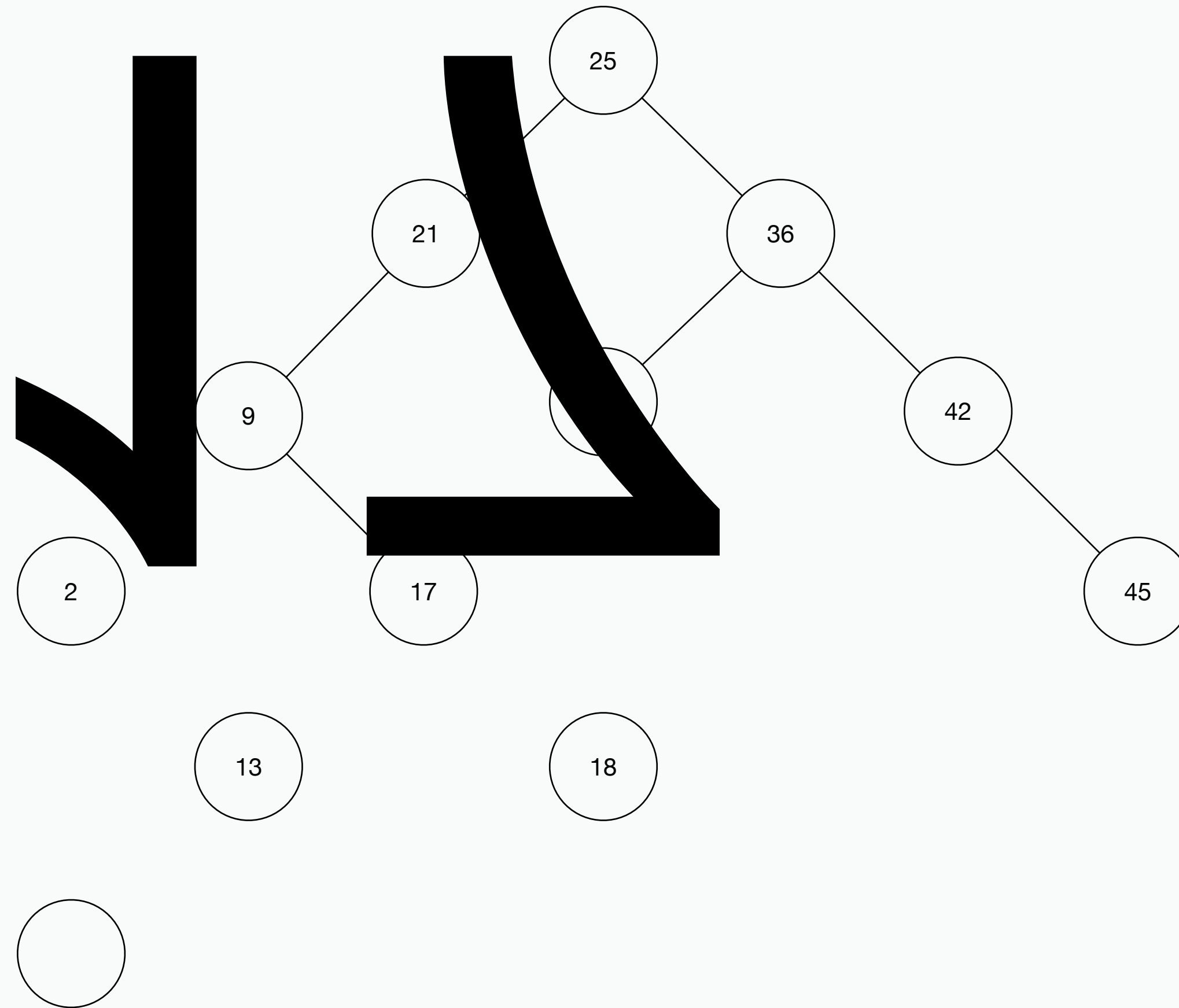
Deleting nodes in a BST

Case 2: Target node has one child.



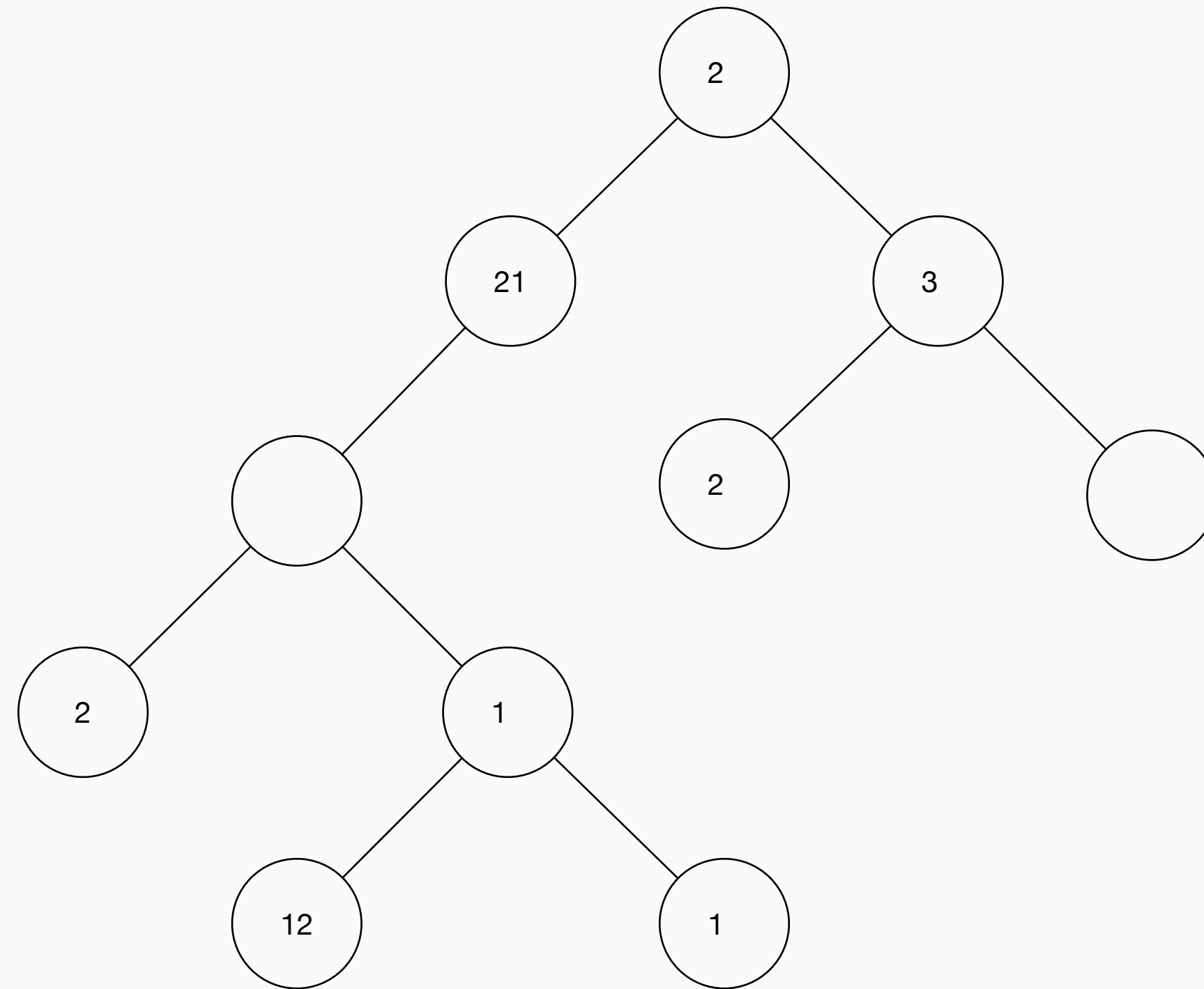
Deleting nodes in a BST

Case 2: Target node has one child.



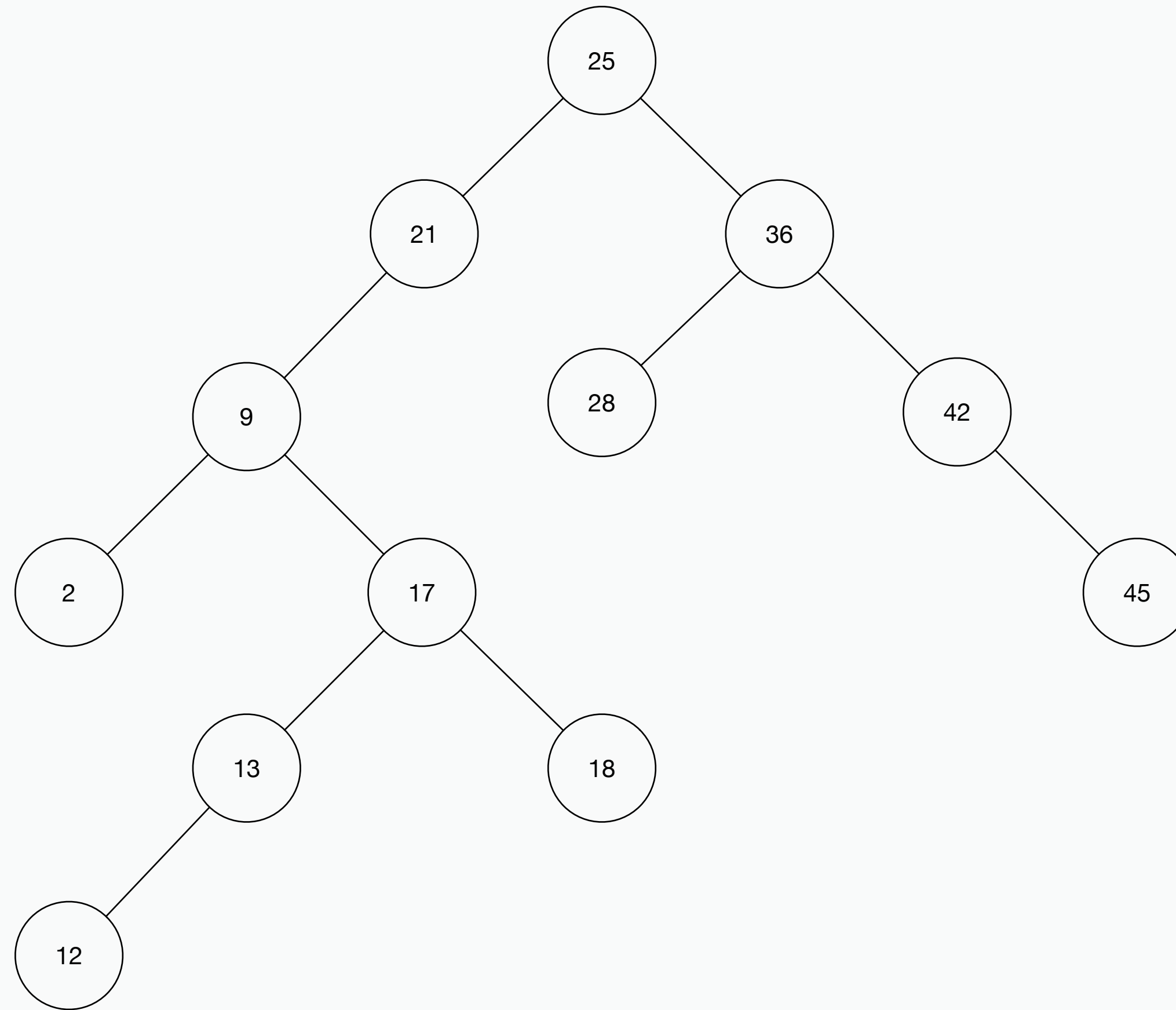
Deleting nodes in a BST

Case 2: Target node has one child.



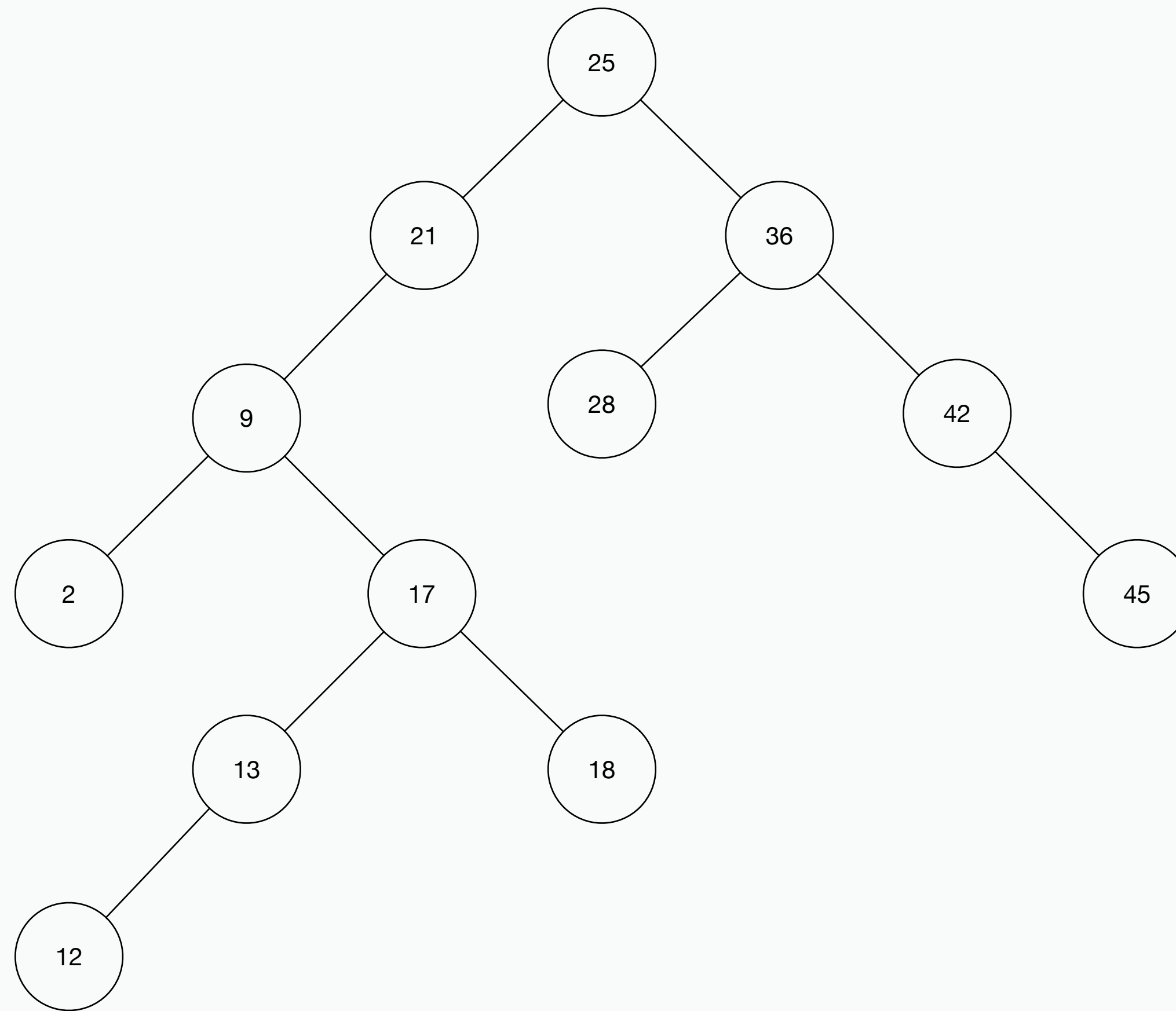
Deleting nodes in a BST

Cases 3 & 4: Target node has two children



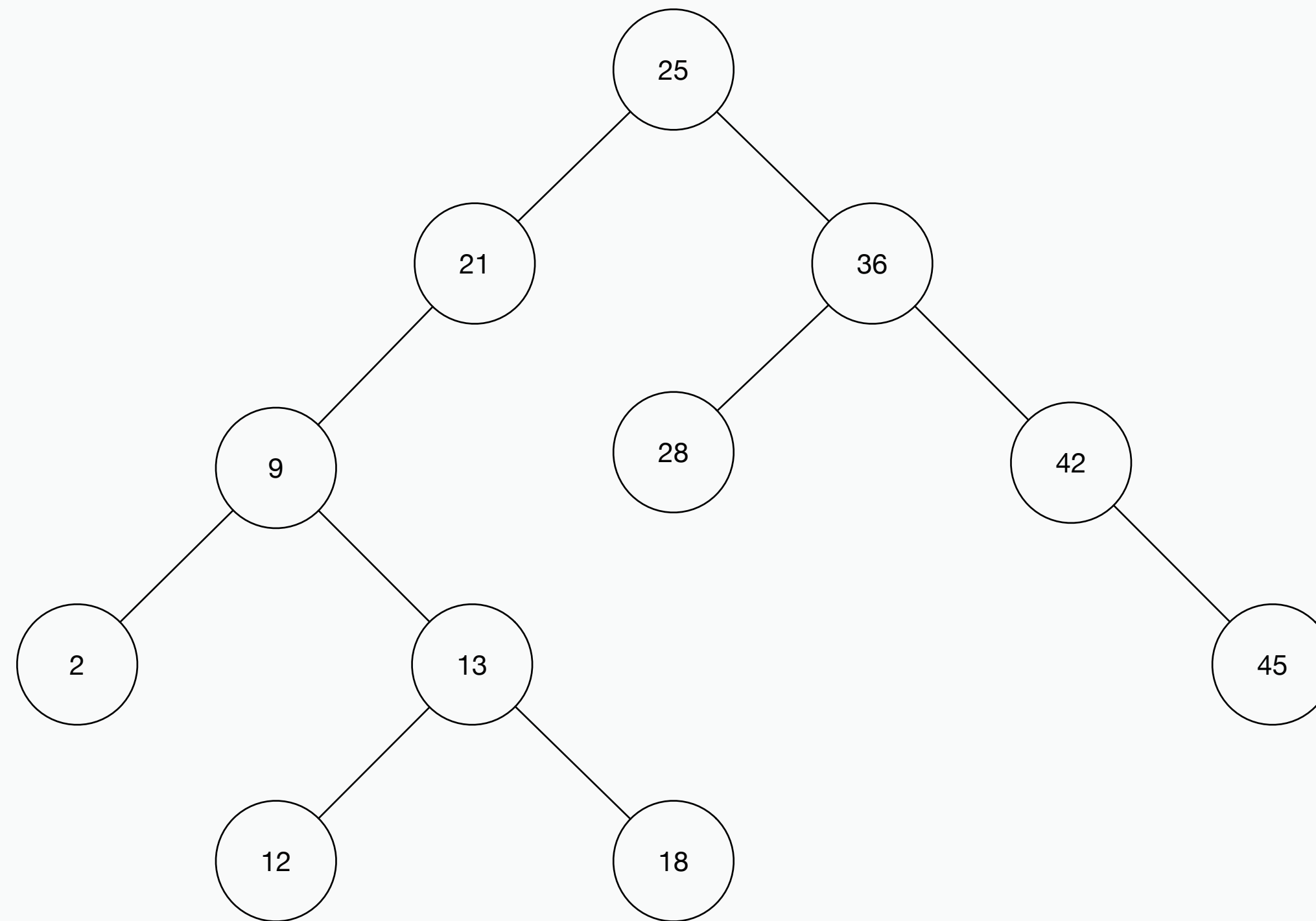
Deleting nodes in a BST

Case 3: Target node has two children, but left child has no right child.



Deleting nodes in a BST

Case 3: Target node has two children, but left child has no right child.

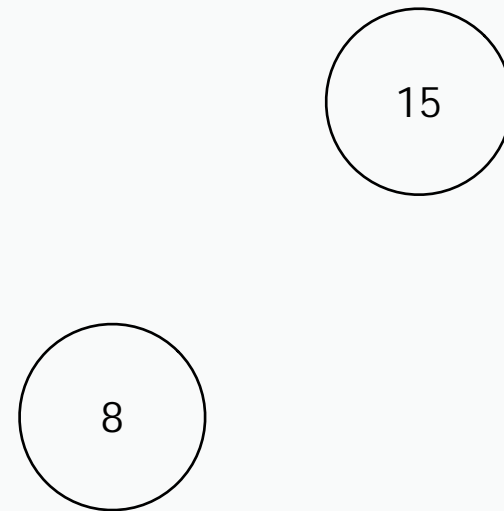


Deleting nodes in a BST

Case 4: Target node has two children, but left child has a right child.

Deleting nodes in a BST

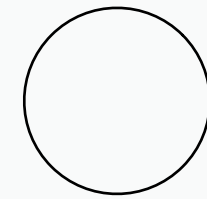
Case 4: Target node has two children, but left child has a right child.



Deleting nodes in a BST

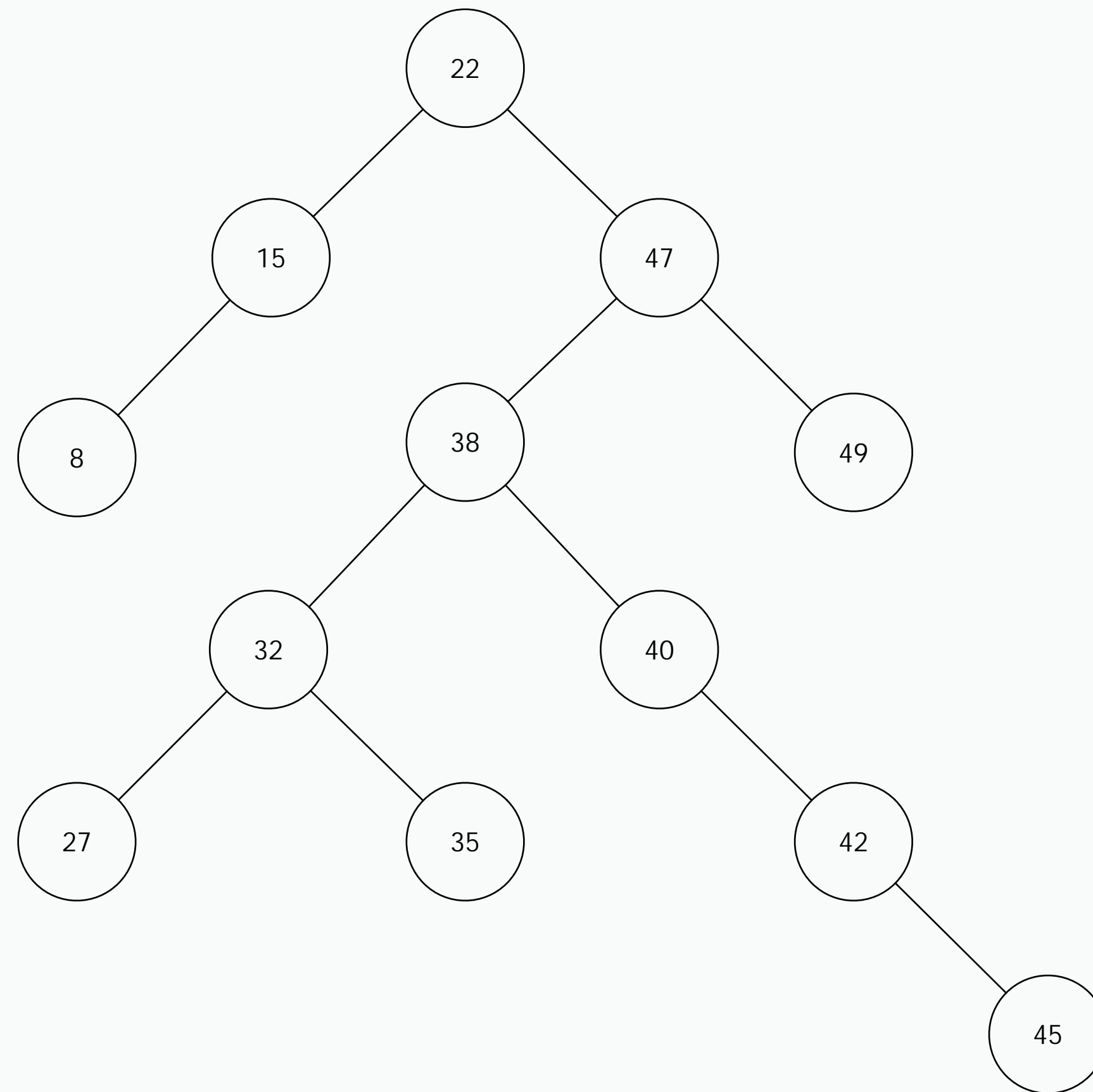
Deleting nodes in a BST

Case 4: Target node has two children, but left child has a right child.



Deleting nodes in a BST

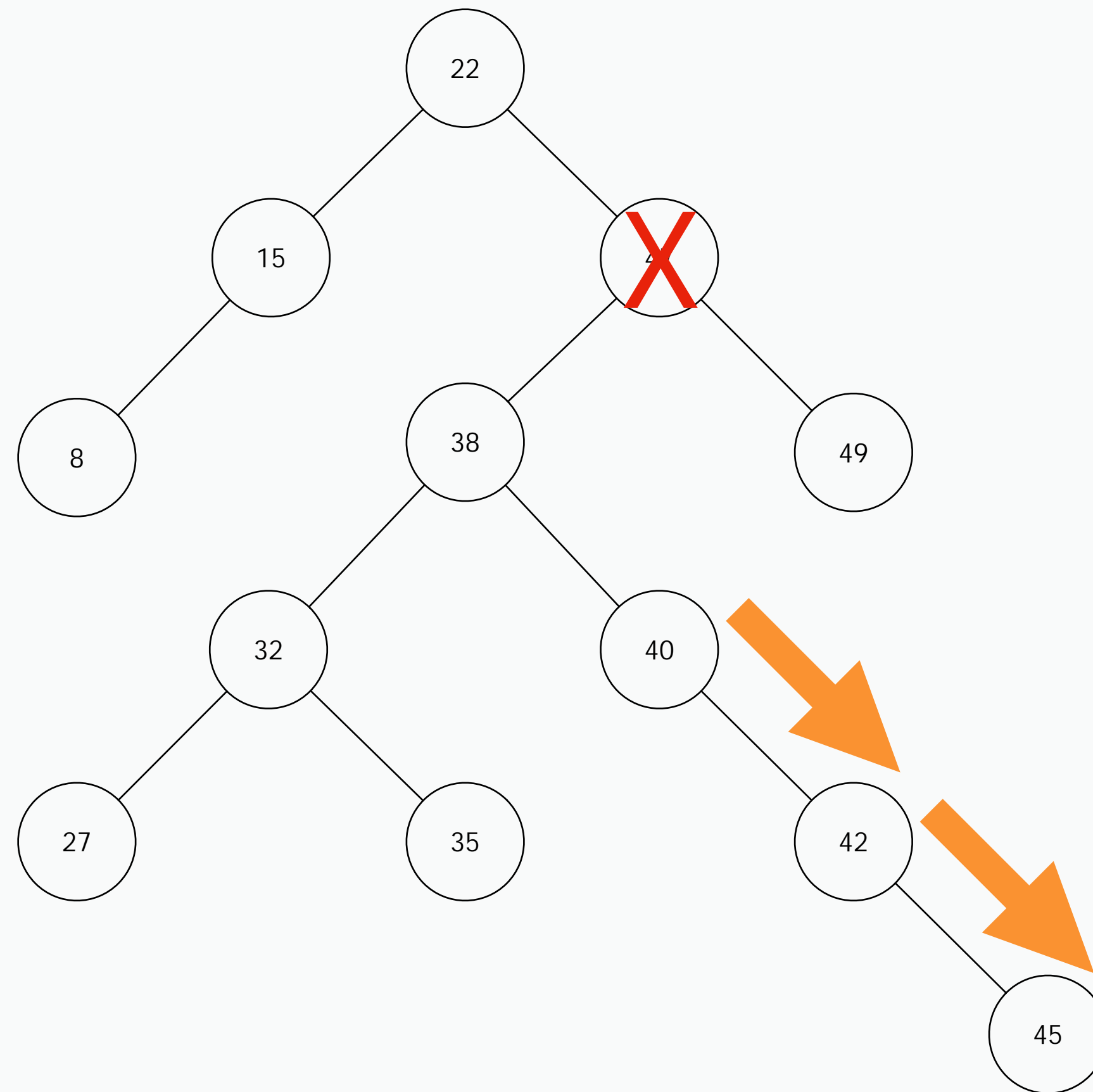
Case 4: Target node has two children, but left child has a right child.



Deleting nodes in a BST

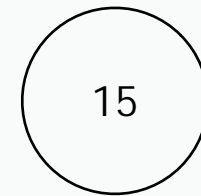
Deleting nodes in a BST

Case 4: Target node has two children, but left child has a right child.



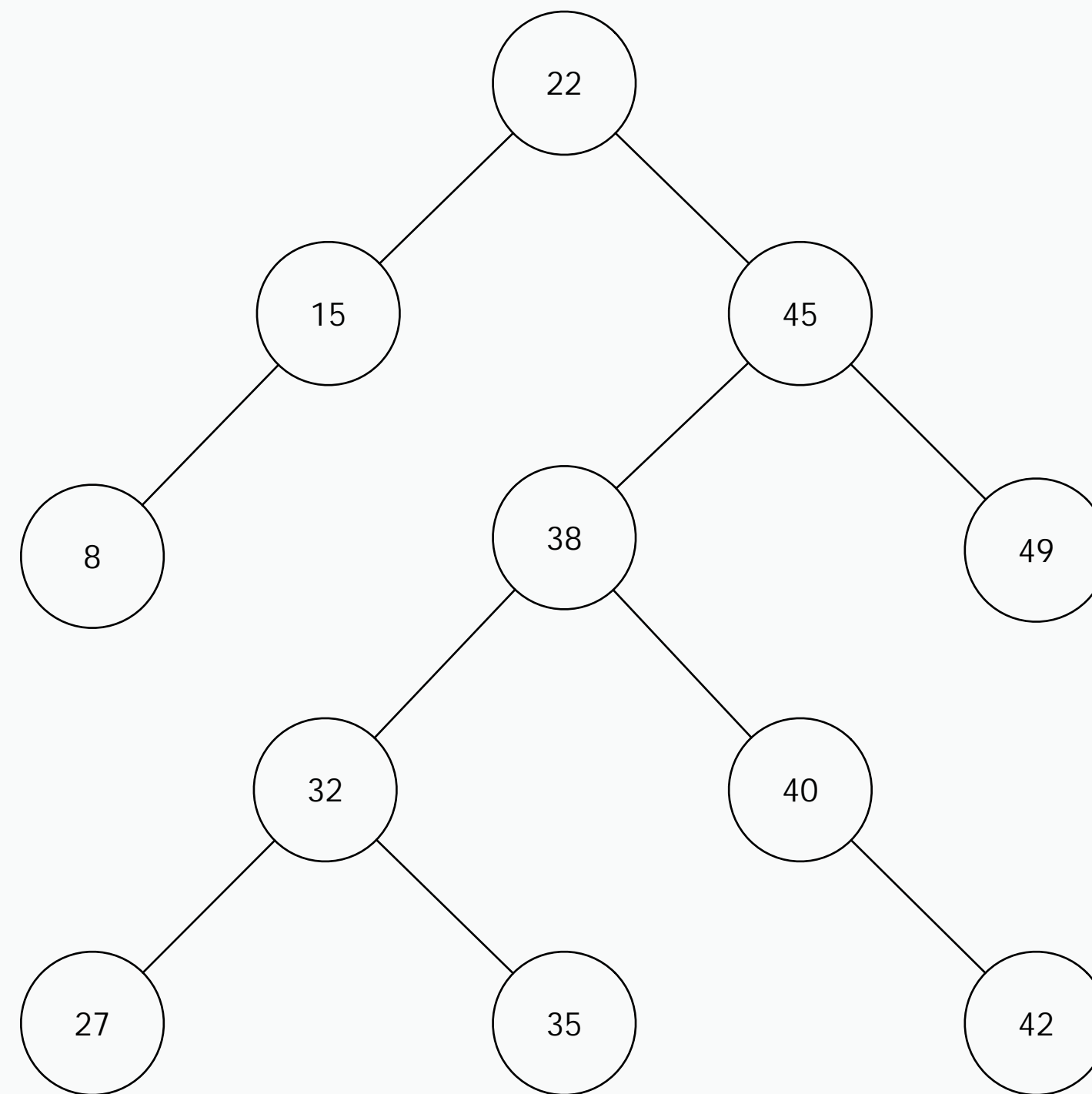
Deleting nodes in a BST

Case 4: Target node has two children, but left child has a right child.



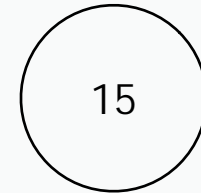
Deleting nodes in a BST

Case 4: Target node has two children, but left child has a right child.



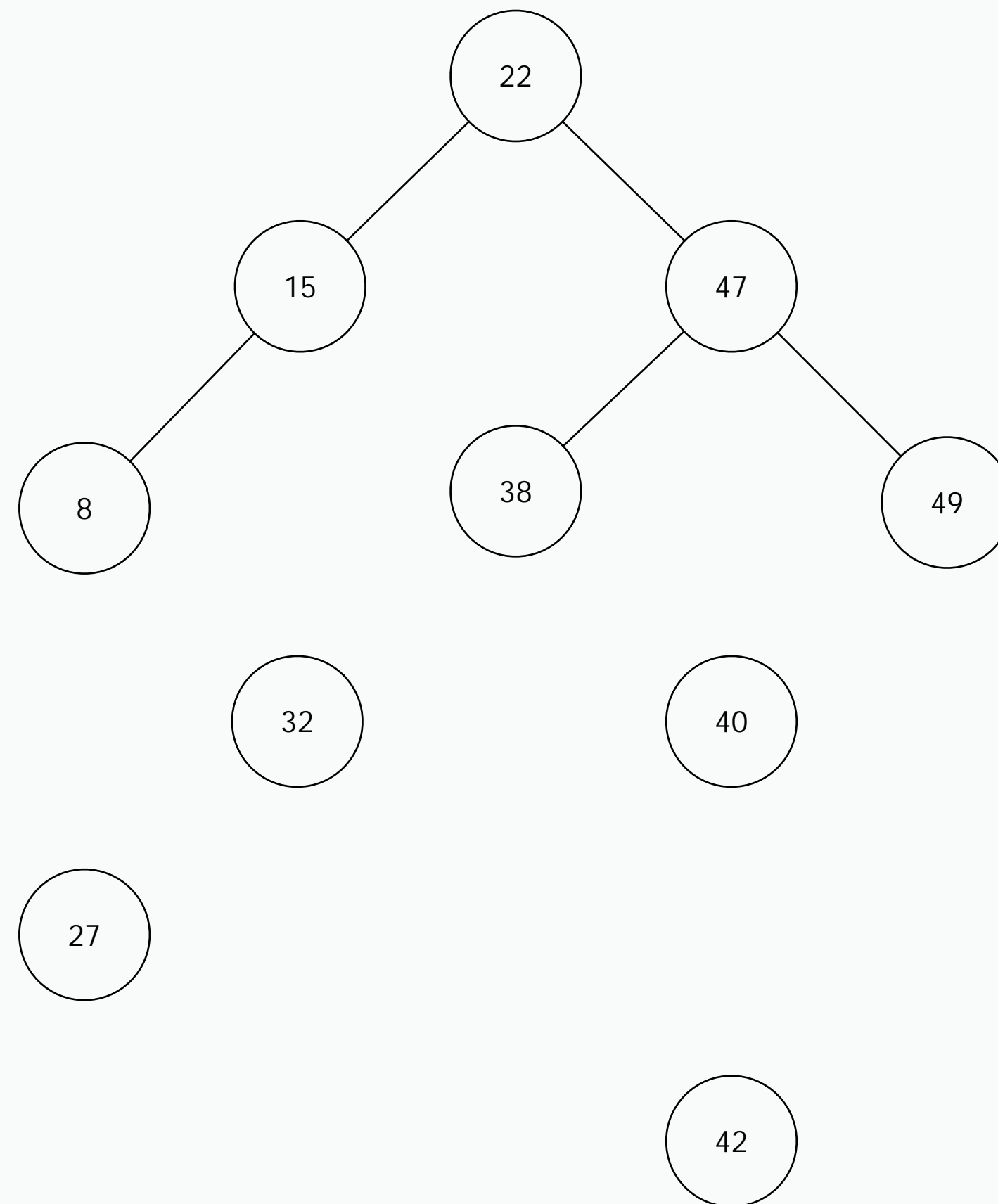
Deleting nodes in a BST

Case 4: Target node has two children, but left child has a right child.



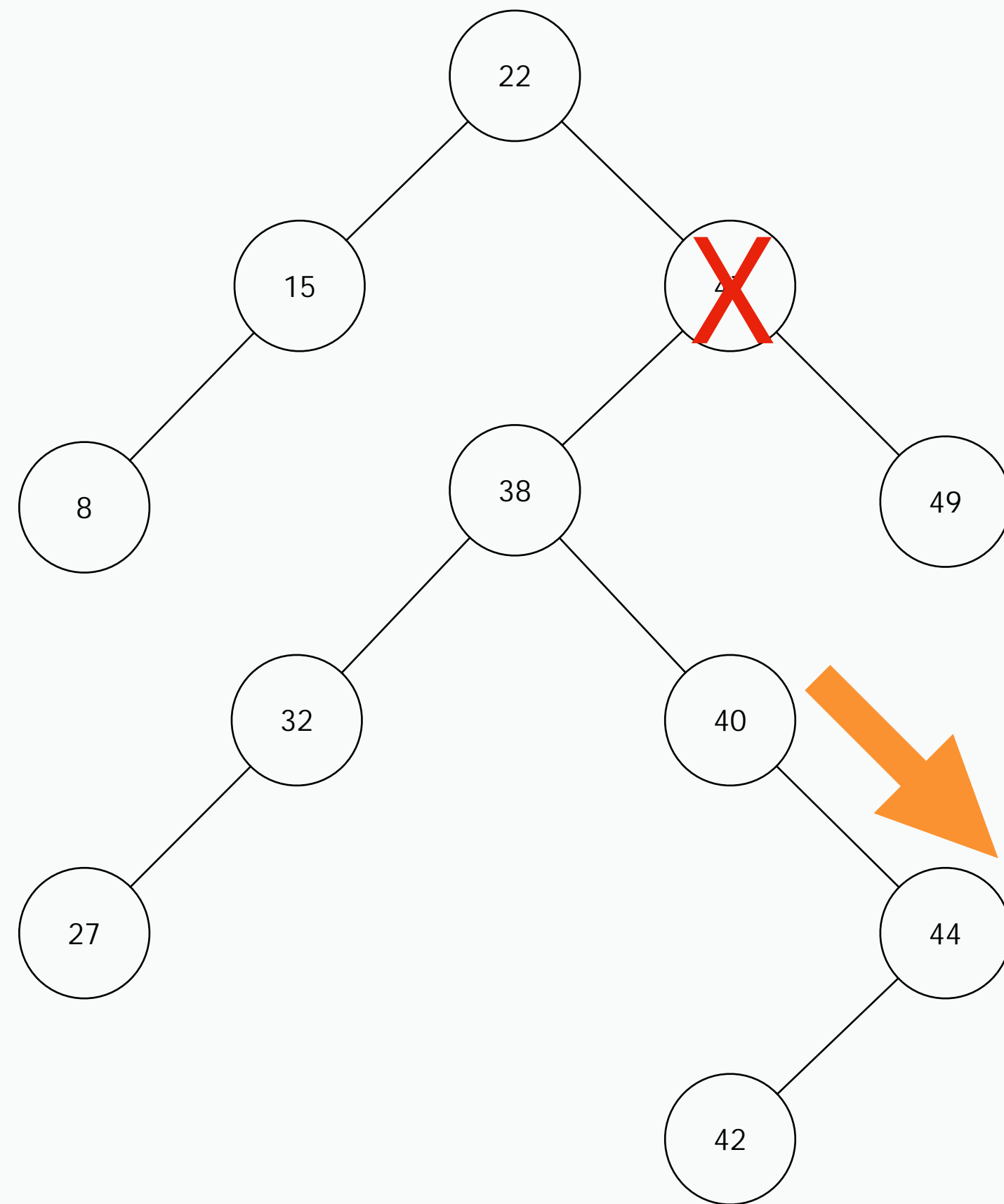
Deleting nodes in a BST

Case 4: Target node has two children, but left child has a right child.



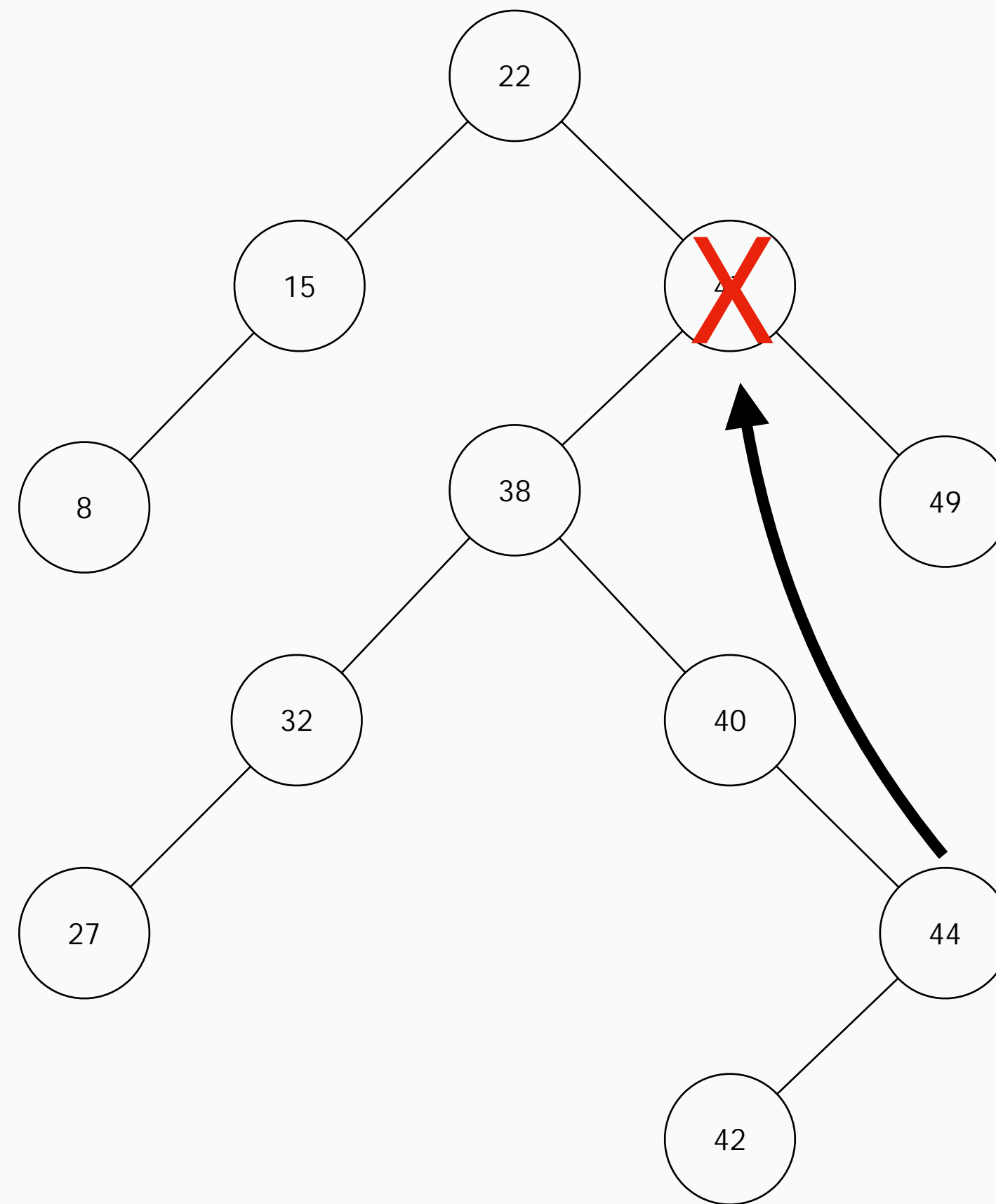
Deleting nodes in a BST

Case 4: Target node has two children, but left child has a right child.



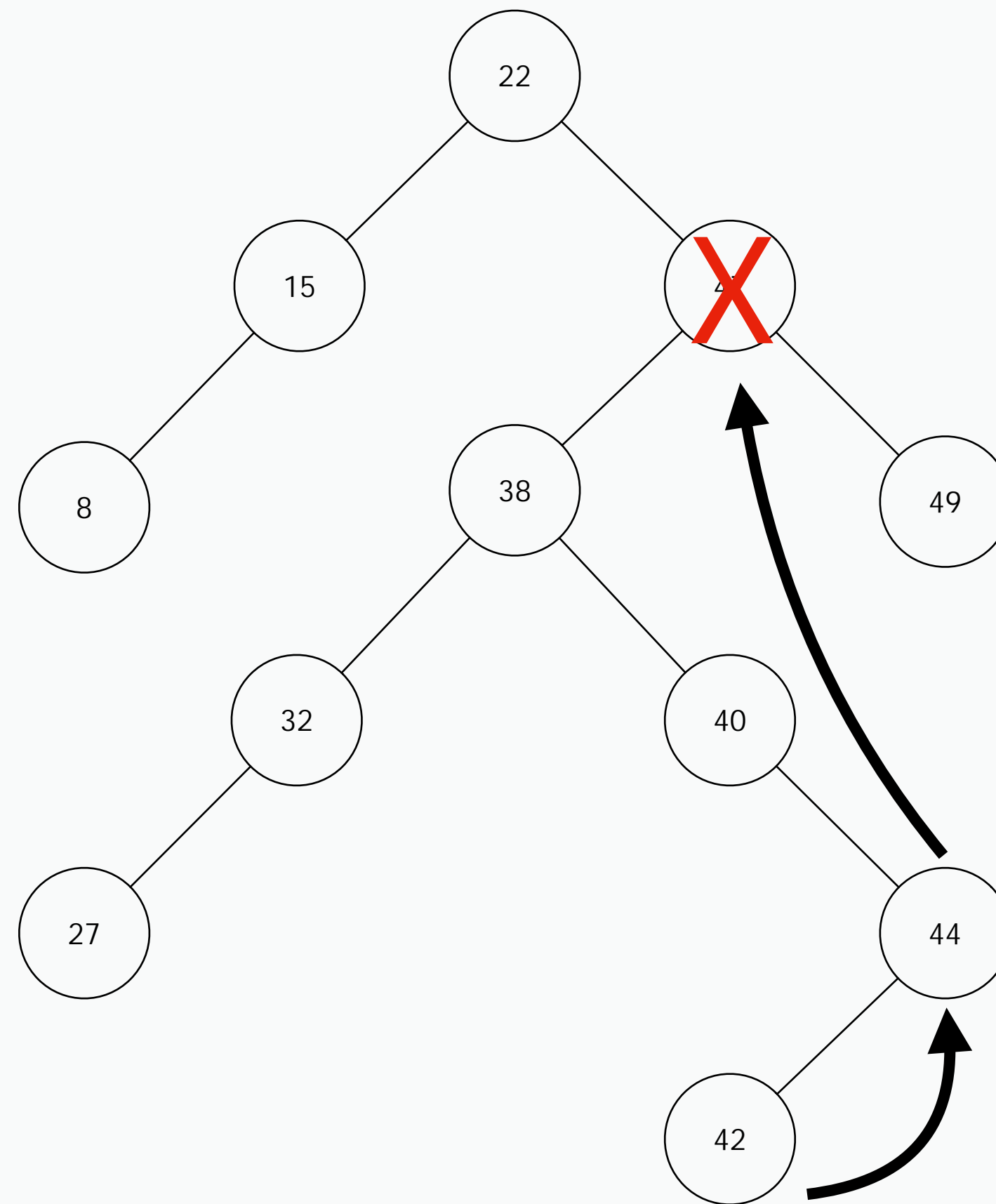
Deleting nodes in a BST

Case 4: Target node has two children, but left child has a right child.



Deleting nodes in a BST

Case 4: Target node has two children, but left child has a right child.



Complexity of BST operations

	Insert node	Search	Delete node
Average case	$O(\log N)$	$O(\log N)$	$O(\log N)$
Worst case	$O(N)$	$O(N)$	$O(N)$